

نمبر سسٹم

لینٹ 2

اس باب کا مطالعہ کرنے کے بعد طلبہ اس قابل ہو جائیں گے وہ:

- مختلف نمبر سسٹم، بشمول اعشاریہ، بائری، ہیکسا ڈسیمیل اور آکٹل اور ان کی متعلقہ بنیادی قدروں اور ہندسوں کو سمجھ سکیں۔
- سمجھ سکیں کہ کمپیوٹر بنیادی طور پر ڈیٹا کی نمائندگی کے لیے بائری نمبر سسٹم کیوں استعمال کرتے ہیں۔
- سمجھ سکیں کہ ڈیٹا کی مشینی سطح کی نمائندگی، بشمول کمپیوٹر کے آرکیٹیکچر میں ڈیٹا کو کیسے ذخیرہ اور پروسیس کیا جاتا ہے۔
- کمپیوٹر میں مکمل اور حقیقی اعداد کی نمائندگی، بشمول دونوں کے لیے بائری انکوڈنگ کے طریقے سمجھ سکیں۔
- سمجھ سکیں کہ اعداد کی بائری نمائندگی پر مختلف ریاضی کے عمل، جیسے کہ جمع، تفریق، ضرب اور تقسیم کیسے انجام پاتے ہیں؟
- جان سکیں کہ عام ٹیکسٹ انکوڈنگ اسکیموں کا تصور، جیسے ASCII اور Unicode اور یہ کیسے حروف کی نمائندگی کرتے ہیں۔
- سمجھ سکیں کہ ڈیجیٹل ڈیٹا کی نمائندگی ملٹی میڈیا کی مختلف شکلوں جیسے کہ تصاویر، آڈیو، ویڈیو اور دیگر ملٹی میڈیا وسائل کے لیے کیسے کام کرتی ہے۔
- سمجھ سکیں کہ مخصوص ایپلی کیشنز کے لیے مختلف فائل فارمیٹس اور ان کے تغیرات کیا ہیں۔
- فائل ایکسٹینشن کا تصور اور فائل کی اقسام اور ایپلی کیشنز کی شناخت میں ان کی اہمیت سمجھ سکیں۔
- ڈیٹا کی نمائندگی سے متعلق کلیدی اصطلاحات، بشمول ASCII، یونیکوڈ، بائری، سائینڈ اور ان سائینڈ نمبر، ٹس، ہائس، ہیکسا ڈسیمیل نمبر سسٹم، بائری میں منفی اعداد، دو کا کمپلیمنٹ، بائری ارتھمیٹک، اوورفلو اور انڈرفلو کو سمجھ سکیں۔
- جان سکیں کہ متغیرات اور بولین آپریٹرز کا استعمال کرتے ہوئے بولین ایکسپریژن کو کیسے بنایا جائے۔
- بولین الجبرا میں ڈوئلٹی کا تصور جان سکیں، جہاں AND، OR، بن جاتا ہے اور 1، 0 بن جاتا ہے۔
- ڈیجیٹل لاجک کے بنیادی اصول، جس میں معلومات کو پروسیس کرنے اور ذخیرہ کرنے کے لیے بائری ہندسوں (0 اور 1) کا استعمال شامل ہے سمجھ سکیں۔
- اینالاگ اور ڈیجیٹل سگنلز کے درمیان فرق اور ان کے اہم فرق کو سمجھ سکیں۔
- مختلف لاجک گیٹز (XOR, NAND, NOT, OR, AND) اور بائری ڈیٹا کی پروسیسنگ میں ان کے افعال سمجھ سکیں۔
- بولین ایکسپریژن کو آسان بنانے کے لیے بصری ٹول کے طور پر کارنائف میپ کا استعمال کر سکیں۔
- ٹرو تھ ٹیبل، بولین ایکسپریژن، ہاف ایڈر اور فل ایڈر کا سرکٹ ڈیاگرام بنا سکیں۔

تعارف

- کمپیوٹر سائنس اور ڈیجیٹل الیکٹرانکس میں نمبر سسٹم کو سمجھنا بنیادی ضرورت ہے۔ اس باب میں مختلف نمبرنگ سسٹمز، ان کی ایپلی کیشنز اور کمپیوٹر میں ان کے استعمال کے بارے میں بات کی جائے گی۔ ہم مندرجہ ذیل موضوعات کا احاطہ کریں گے:
- مختلف نمبرنگ سسٹم: اعشاریہ، بائری، ہیکسا ڈسیمیل اور آکٹل۔
 - کمپیوٹر میں بائری نمبر سسٹم

- مشین کی سطح پر ڈیٹا کی نمائندگی
- مکمل اور حقیقی اعداد کی نمائندگی
- بائنری حسابی عملیات
- ٹیکسٹ انکوڈنگ اسکیمیں: آسکی اور یونیکوڈ
- فائل فارمیٹس اور ایکسٹینشنز
- ڈیٹا کی نمائندگی میں اہم اصلاحات
- بائنری ڈیٹا میں رد و بدل
- فائل فارمیٹس کے درمیان اختلافات
- کمپیوٹرز میں تصاویر، آڈیو اور ویڈیو ذخیرہ کرنا

2.1 نمبرنگ سسٹم (Numbering System)

کمپیوٹنگ میں نمبرنگ سسٹم ضروری ہیں کیونکہ وہ ڈیٹا کی نمائندگی، ذخیرہ اور پروسیسنگ کی بنیاد فراہم کرتے ہیں۔ مختلف نمبرنگ سسٹم کمپیوٹرز کو حساب کتاب، ڈیٹا اسٹوریج اور ڈیٹا ٹرانسفر جیسے کام انجام دینے میں مدد کرتے ہیں۔ یہ نظام کمپیوٹرز کو مختلف قسم کی معلومات کی نمائندگی کرنے کی اجازت دیتے ہیں، جیسے ٹیکسٹ، رنگ اور میموری ایڈریس وغیرہ۔ یہاں چند نمبرنگ سسٹم کی وضاحت ہے:

2.1.1 اعشاریہ نظام

اعشاریہ ریڈیسیمل نمبر سسٹم دس (10) کی بنیاد پر قائم نمبر سسٹم ہے جو 0 سے 9 تک کے ہندسوں پر مشتمل ہوتا ہے۔ ہم روزمرہ زندگی میں اکثر اس نمبر سسٹم کا استعمال کرتے ہیں۔ اس نمبر سسٹم میں ہر ہندسہ 10 کی طاقت کی نمائندگی کرتا ہے۔ اعشاریہ نظام میں دائیں سے شروع ہونے والے مقامات کی قدریں 10 کی طاقتیں ہوتی ہیں جیسے: 10^0 ، 10^1 ، 10^2 وغیرہ۔ مثال کے طور پر اعشاریہ نمبر 523 کا مطلب یہ ہے:

$$5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 = 500 + 20 + 3 = 523$$

2.1.2 بائنری سسٹم

بائنری میں جگہ کی قدریں دائیں طرف سے واقع ہوتی ہیں، 2^0 سے شروع ہوتی ہیں اور 2^n پر ختم ہوتی ہیں۔ یہاں ہر پوزیشن 2 کی پاور کو ظاہر کرتی ہے۔

مثال کے طور پر، بائنری نمبر 1011 کو درج ذیل انداز میں اعشاریہ نمبر میں تبدیل کیا جاسکتا ہے:

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11_{10}$$

کمپیوٹر بانسری سسٹم میں ہی کام کرتے ہیں کیونکہ یہ طریقہ الیکٹرانکس کے لیے مناسب ہے۔ ڈیجیٹل سرکٹس کی دو حالتیں ہیں: ”یہ آن یا آف ہو سکتے ہیں۔“ ان حالتوں کو بانسری ہندسوں کے ذریعے بہت آسانی سے ظاہر کیا جاسکتا ہے۔ 1 آن کو جب کہ 0 آف کو ظاہر کرتا ہے۔ جب آپ کی بورڈ پر ٹائپ کرتے ہیں تو کمپیوٹر ہر حرف کا بانسری میں ترجمہ کرتا ہے۔ اسی طرح نمبر متن، تصاویر اور آواز اپنی کم ترین سطح پر بانسری شکل میں تبدیل ہو جاتے ہیں۔ اس پر ہم بعد میں اس باب میں تفصیل سے بحث کریں گے۔ جب آپ اپنے کی بورڈ پر خط ٹائپ کرتے ہیں تو، کمپیوٹر اسے بانسری کوڈ میں تبدیل کرتا ہے۔ اسی طرح، تمام قسم کے اعداد و شمار، بشمول نمبر، متن، تصاویر اور آوازیں، بالآخر بانسری کوڈ میں تبدیل ہوتے ہیں۔ اسی باب میں آگے جا کر اس کا مزید جائزہ لیں گے۔

2.1.2.1 اعشاریہ سے بانسری میں تبدیلی

- مندرجہ ذیل الگورتھم اعشاریہ نمبر کو بانسری میں تبدیل کرتا ہے۔
- اعشاریہ نمبر کو بانسری شکل میں تبدیل کرنے کے لیے اعشاریہ نمبر کو 2 پر تقسیم کریں۔
- بقیہ لکھیں۔
- نمبر کو 2 سے تقسیم کریں یہاں تک کہ تقسیم کے بعد جو حصہ باقی رہ جاتا ہے وہ 0 ہو جائے۔
- مطلب یہ ہے کہ بانسری نمبر کی نمائندگی بقیہ حصے کے ذریعے کی جاتی ہے اور اسے نیچے سے اوپر کی طرف پڑھا جاتا ہے۔

مثال: 83 کو بانسری میں تبدیل کریں۔

$$83 / 2 = 41 \text{ بقیہ } 1$$

$$41 / 2 = 20 \text{ بقیہ } 1$$

$$20 / 2 = 10 \text{ بقیہ } 0$$

$$10 / 2 = 5 \text{ بقیہ } 0$$

$$5 / 2 = 2 \text{ بقیہ } 1$$

$$2 / 2 = 1 \text{ بقیہ } 0$$

$$1 / 2 = 0 \text{ بقیہ } 1$$

مندرجہ بالا اقدامات گرافیکل طور پر شکل 2.1 میں دکھائے گئے ہیں۔ اگر بقیہ کو نیچے سے اوپر کی

طرف پڑھا جائے تو یہ بانسری میں مطلوبہ نتیجہ دیتا ہے، جو 1010011 ہے۔

2	83
2	41-1
2	20-1
2	10-0
2	5-0
2	2-1
2	1-0

سرگرمی

• حاصل کردہ نمبرز کی تبدیلی:

ہر طالب علم ہر مضمون کے لیے آٹھویں جماعت میں حاصل کردہ نمبر لے گا اور انھیں اعشاریہ سے بائری میں تبدیل کرے گا۔ مثال کے طور پر اگر کوئی طالب علم ریاضی میں 85 نمبر حاصل کرتا ہے تو وہ 85 کو بائری میں تبدیل کرے گا (جو کہ 1010101_2 ہے)۔

• گھڑی کا وقت تبدیل کرنا:

طلبہ کو دن کے مختلف اوقات دیے جائیں گے اور انھیں بائری میں تبدیل کرنے کے لیے کہا جائے گا۔ مثال کے طور پر 3:45 (بچے) کو مندرجہ ذیل طور پر تبدیل کیا جائے گا:

$$101101 = (45) \text{ منٹ} \quad 1111 = (15) \text{ گھنٹے}$$

• اپنے سونے کا وقت بائری میں لکھیں۔

2.1.3 آکٹل سسٹم

آکٹل سسٹم ایک عددی نظام ہے جس کی بنیاد 8 ہے، یعنی استعمال ہونے والے ہندسے 0 سے 7 تک ہوتے ہیں۔ آخری ہندسہ 8 کی طاقت کو ظاہر کرتا ہے، جبکہ دیگر ہندسے اس کے مضروب (Coefficient) ہوتے ہیں۔ آکٹل نظام میں، جگہ کی قدریں $8^0, 8^1, 8^2$ وغیرہ ہوتی ہیں۔

مثال کے طور پر، آکٹل نمبر 157 کا مطلب ہے:

$$1 \times 8^2 + 5 \times 8^1 + 7 \times 8^0 = 64 + 40 + 7 = 111_{10}$$

ہر آکٹل ہندسہ تین بائری ہندسوں (بٹس) کی نمائندگی کرتا ہے کیونکہ آکٹل سسٹم کی بیس 8 ہے اور بائری سسٹم کی بیس 2 ہے۔ یہ رشتہ اس حقیقت سے پیدا ہوتا ہے کہ 2 کی پاور 3 یعنی 2^3 کا مطلب ہے 8۔ لہذا ہر آکٹل ہندسے کو تین بائری ہندسوں (بٹس) کے ذریعے پیش کیا جاسکتا ہے۔ اس کا مطلب یہ ہے کہ آکٹل میں 0 سے 7 تک کی کسی بھی قدر کو 3 بٹ بائری نمبر میں تبدیل کیا جاسکتا ہے۔ یہ تعلق بائری اور آکٹل کے درمیان تبدیلی کو سیدھا اور مؤثر بناتا ہے۔

مثیل 2.1 آکٹل اور بائری ہندسوں کے درمیان تعلق کو ظاہر کرتا ہے۔ 110101011_2 (9 بٹ) بائری نمبر پر غور کریں۔ اس نمبر کو

تین گروہوں میں تقسیم کیا جاسکتا ہے:

دائیں سے بائیں تک کے ٹکڑے:

تین بٹس کا ہر گروپ ایک آکٹل ہندسے سے مطابقت رکھتا ہے:

$$110 = 6$$

$$101 = 5$$

$$011 = 3$$

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

مثیل 2.1: آکٹل اور بائری ہندسوں کے درمیان مطابقت

لہذا 110101011 بائری نمبر آکٹل میں 653 کے برابر ہے۔ اگر ایک بائری نمبر کو تین بٹس کے گروپوں میں تقسیم نہیں کیا جاسکتا تو اسے مکمل کرنے کے لیے بائیں طرف صفر شامل کریں۔

نوٹ کریں کہ آکٹل نمبر سسٹم اصل میں جدید کمپیوٹرز میں استعمال نہیں کیا جاتا ہے۔ لہذا ہم یہ کہہ سکتے ہیں کہ 110101011 بائری نمبر آکٹل میں 653 کے برابر ہے۔ جب بھی آپ کے پاس ایک بائری نمبر ہوتا ہے جسے تین کے گروپوں میں تقسیم نہیں کیا جاسکتا ہے تو آپ کو اسے مناسب بنانے کے لیے اس کے بائیں سرے تک صفر شامل کرنا ہوگا۔

2.1.3.1 اعشاریہ سے آکٹل میں تبدیلی

ذیل میں دیا گیا الگورتھم اعشاریہ نمبر کو آکٹل میں تبدیل کرتا ہے۔

- اعشاریہ نمبر کو مساوی حصے میں تبدیل کرنے کے لیے نمبر کو 8 سے تقسیم کریں۔
- بقیہ رقم لکھیں۔

اس کے بعد حاصل شدہ حصے کو 8 سے دوبارہ تقسیم کریں۔

• تقسیم کو جاری رکھیں جب تک کہ نمبروں میں سے ایک کا نتیجہ 0 نہ نکل جائے۔

• آکٹل کی بیس آٹھ ہے اور یہ نیچے سے اوپر تک پڑھا جانے والا بقیہ نمبر ہے۔

مثال کے طور پر: 83 کو آکٹل میں تبدیل کریں۔

$$83 / 8 = 10 \text{ بقیہ } 3$$

$$10 / 8 = 1 \text{ بقیہ } 2$$

$$1 / 8 = 0 \text{ بقیہ } 1$$

مندرجہ بالا اقدامات گرافیکل طور پر شکل 2.2 میں دکھائے گئے ہیں۔ نیچے سے اوپر جاتے ہوئے، بقیہ پڑھنے سے مطلوبہ نتیجہ ملے گا، جو آکٹل سسٹم میں 123 ہے۔

سرگرمی

- مندرجہ ذیل ڈیسمل نمبروں کو آکٹل میں تبدیل کرنے کے لیے جوڑیوں میں کام کریں:

45، 128، 64

- ان آکٹل نمبروں کو ڈیسمل میں تبدیل کریں:

57، 124، 301

- کلاس کے ساتھ اپنے جوابات کا اشتراک کریں اور کسی بھی اختلافات پر تبادلہ خیال کریں۔

کیا آپ جانتے ہیں؟

ابتدائی کمپیوٹنگ سسٹم جیسے پی ڈی پی-8 میں آکٹل سسٹم استعمال کیا گیا تھا۔ یہ استعمال کیا گیا تھا کیونکہ اعشاریہ اور بائنری کے مقابلے میں آکٹل اور بائنری کے درمیان تبدیل کرنا آسان ہے۔

دلچسپ معلومات:

نمبر سسٹم کے درمیان تبدیل کرتے وقت، درستگی کو یقینی بنانے کے لیے اپنے بقیہ اور رقم کو ڈبل چیک کریں۔ تبادلے کے عمل کے ساتھ زیادہ آرام دہ بننے کے لیے مختلف نمبروں کے ساتھ مشق کریں۔

2.1.4 ہیکسا ڈسیمیل سسٹم (Hexadecimal System)

ہیکسا ڈسیمیل ایک 16 بیس نمبر سسٹم ہے جس میں 0 سے 9 اور A سے F حروف تہجی ہیں۔ ہر ہندسہ 16 کی پاور کی نمائندگی کرتا ہے جس سے اس ہندسہ کی پوزیشن واضح ہوتی ہے۔ حرف A سے F کا مطلب 10 سے 15 کی عددی قدر ہے۔ ہیکسا ڈسیمیل میں ہندسے دائیں سے بائیں کی طرف جاتے ہیں جو $16^0, 16^1, 16^2$ وغیرہ ہیں۔ مثال کے طور پر ہیکسا ڈسیمیل نمبر 1A3 کو اعشاریہ میں اس طرح ظاہر کیا جاسکتا ہے:

$$1 \times 16^2 + A \times 16^1 + 3 \times 16^0 = 1 \times 256 + 10 \times 16 + 3 \times 1 = 256 + 160 + 3 = 419_{10}$$

ہیکسا ڈسیمیل نمبر سسٹم براہ راست کمپیوٹرز استعمال نہیں کیا جاتا ہے۔ تاہم یہ آکٹل سے بھی زیادہ ٹھوس نمائندگی فراہم کرتا ہے۔ اس سے ہمارے لیے بڑے بائنری نمبر پڑھنا اور لکھنا آسان ہو جاتا ہے۔

اس کی وجہ یہ ہے کہ ہیکسا ڈسیمیل سسٹم کی بنیاد 16 ہے اور بائنری سسٹم کی بنیاد 2 ہے، اس لیے ہر ہیکسا ڈسیمیل ہندسہ 4 بائنری بٹس کے برابر ہوتا ہے۔ یہ تعلق اس حقیقت سے نکلتا ہے کہ $2^4 = 16$ مطلب یہ ہے کہ اگر آپ کسی بھی ہیکسا ڈسیمیل نمبر کو 0 سے 15 کے درمیان تبدیل کرنا چاہتے ہیں تو اسے 4 بٹ بائنری نمبر میں تبدیل کیا جائے گا۔

ٹیبل 2.2 ہیکسا ڈسیمیل ہندسوں کو بائنری ہندسوں میں تبدیل کرنے کی وضاحت کرتا ہے۔ چار بٹس کا ہر گروپ ایک ہیکسا ڈسیمیل ہندسے کے مساوی ہوتا ہے:

مثال:

لہذا اس معاملے میں بائنری نمبر 1101011010110010، ہیکسا ڈسیمیل نمبر D6B2 کے برابر ہے۔ اگر ایک بائنری نمبر کو 4 بٹس کے طور پر گروپ نہیں کیا جاسکتا ہے تو اسے مکمل کرنے کے لیے نمبر کے سامنے صفر لگائیں۔

1101 0110 1011 0010

Hexadecimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

ٹیبیل 2.2 ہیکسا ڈیسیمل اور بائری نمبر کی مطابقت

$$1101 = D$$

$$0110 = 6$$

$$1011 = B$$

$$0010 = 2$$

2.1.4.1 اعشاریہ کو ہیکسا ڈیسیمل میں تبدیل کرنا

مندرجہ ذیل الگورتھم اعشاریہ نمبر کو ہیکسا ڈیسیمل میں تبدیل کرتا ہے:

- نمبر کو 16 سے تقسیم کر کے اس اعشاریہ نمبر کو مطلق قدر میں تبدیل کریں۔
- حاصل باقی اور کوئٹینٹ (qoutient) کو لکھیں۔
- اس نمبر کو 16 سے تقسیم کرتے رہیں اور کوئٹینٹ (qoutient) کو اس وقت تک لکھتے رہیں جب تک کہ کوئٹینٹ (qoutient) صفر نہ ہو جائے۔
- جیسا کہ آپ نے اندازہ لگایا ہوگا، ہیکسا ڈیسیمل نمبر کے لیے حاصل باقی کو نیچے سے اوپر تک پڑھا جاتا ہے۔

16	2297	
	143 - 9	⇒ 9
	8 - 15	⇒ F
	0 - 8	⇒ 8 ↑

شکل 2.3: اعشاریہ سے ہیکسا ڈیسیمیل

مثال: 2297 کو ہیکسا ڈیسیمیل میں تبدیل کریں۔

$$2297 / 16 = 143 \text{ بقیہ } 9$$

$$143 / 16 = 8 \text{ بقیہ } F$$

$$8 / 16 = 0 \text{ بقیہ } 8$$

مندرجہ بالا اقدامات گرافیکل طور پر شکل 2.3 میں دکھائے گئے ہیں۔ بقیہ کو نیچے سے اوپر تک پڑھنے سے مطلوبہ نتیجہ ملتا ہے، یعنی ہیکسا ڈیسیمیل میں 8 F 9۔

سرگرمی

مندرجہ ذیل اقدار کو ہیکسا ڈیسیمیل میں ظاہر کریں۔ اپنے نتائج اپنے ہم جماعتوں کے ساتھ شیئر اور بحث کریں:

- ووٹ ڈالنے کے لئے کم از کم عمر
- دریائے سندھ کی لمبائی
- پاکستان میں کل اضلاع کی تعداد
- K2 کی اونچائی (دنیا کا دوسرا بلند ترین پہاڑ)
- پاکستان کا رقبہ

2.2 کمپیوٹنگ سسٹم میں ڈیٹا کو ظاہر کرنا

کمپیوٹر بہت ساری معلومات کو پروسیس اور اسٹور کر سکتے ہیں۔ مندرجہ ذیل سیکشن میں عددی (Numeric) ڈیٹا کی نمائندگی پر تبادلہ خیال کریں گے۔

2.2.1 صحیح اعداد (Z) اور حقیقی اعداد (R) کی بائنری انکوڈنگ

جب ہم کمپیوٹرز میں ڈیٹا اسٹور کرتے ہیں، خاص طور پر نمبر، تو یہ سمجھنا ضروری ہے کہ وہ میموری میں کس طرح ظاہر اور محفوظ کیا جاتا ہے۔ آئیے دریافت کرتے ہیں کہ 1، 2 اور 4 بائٹس میں مکمل اعداد کی قدروں کے مختلف سائز کو کس طرح اسٹور کیا جاتا ہے اور ہم مثبت اور منفی عدد کو کس طرح منظم کر سکتے ہیں۔

2.2.2 مکمل اعداد (W) اور صحیح اعداد (Z)

انٹیجر (Integer) صحیح اعداد کے طور پر بھی جانا جاتا ہے۔ ریاضی اور کمپیوٹر سائنس میں اہم عناصر ہیں۔ پروگرامنگ اور ڈیٹا کے ڈیزائن کے ذریعے حل تلاش کرنے والے بہت سے کمپیوٹیشنز کے لیے ان تصورات کا علم اہم ہے۔

2.2.2.1 مکمل اعداد (W)

مکمل اعداد غیر منفی نمبروں کا ایک مجموعہ ہیں جن میں صرف انٹیجر (Integer) ہوتے ہیں۔ یہ صفر ہیں اور تمام مثبت عدد ہیں۔ ریاضیاتی طور پر مکمل اعداد کا مجموعہ یہ ہے:

$$W = \{0, 1, 2, 3, \dots\}$$

کمپیوٹنگ میں مکمل اعداد اکثر ایسی مقدار کو ظاہر کرنے کے لیے استعمال ہوتے ہیں جو منفی نہیں ہو سکتے ہیں۔ مثال کے طور پر ایک اسکول میں طالب علموں کی تعداد، کسی شخص کی عمر اور گریڈ جہاں کریڈٹ پوائنٹ منفی نہیں ہیں۔

ایک بائٹ کے صحیح عدد میں قدروں کو اسٹور کرنے کے لیے 8 بٹس ہوتے ہیں۔ اگر تمام 8 بٹس آن ہیں تو یہ زیادہ سے زیادہ قیمت کی نمائندگی کرتا ہے 1111111_2 جو 255_{10} ہے۔ اگر تمام بٹس آف ہیں تو یہ کم از کم قیمت 00000000_2 کی نمائندگی کرتا ہے جو 0_{10} ہے۔ اسی طرح، اگر ہم 2 یا 4 بائٹس استعمال کرتے ہیں تو ہمیں ڈیٹا اسٹور کرنے اور بڑی اقدار کو ذخیرہ کرنے کے قابل ہونے کے لیے مزید بٹس ملتے ہیں۔ اگر بٹس کی تعداد n ہے تو زیادہ سے زیادہ قیمت جس کی ہم نمائندگی کر سکتے ہیں وہ $2^n - 1$ ہے۔ اسی طرح:

- 1- بائٹ مکمل عدد (8 بٹس): زیادہ سے زیادہ قیمت $2^8 - 1 = 255$
- 2- بائٹ مکمل عدد (16 بٹس): زیادہ سے زیادہ قیمت $2^{16} - 1 = 65535$
- 3- بائٹ مکمل عدد (32 بٹس): زیادہ سے زیادہ قیمت $2^{32} - 1 = 4,294,967,295$

2.2.2.2 انٹیجر (Z)

انٹیجر منفی نمبروں کو شامل کرنے کے لیے مکمل اعداد کے تصور کو بڑھاتے ہیں۔ کمپیوٹر پروگرامنگ کی دنیا میں ہم انھیں سائنڈ انٹیجر کہتے ہیں۔ انٹیجر اعداد کا مجموعہ اس طرح پیش کیا جاتا ہے:

$$Z = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$$

اس میں مثبت اور منفی دونوں اقدار کو اسٹور کرنے کے لیے ایک بٹ کو سائن بٹ (سب سے اہم بٹ) کے طور پر محفوظ کیا جاتا ہے۔ اگر سائن بٹ آن ہے تو، قیمت منفی ہے۔ بصورت دیگر، یہ مثبت ہے۔ اس نظام کا استعمال کرتے ہوئے، زیادہ سے زیادہ مثبت قدر جو 1 بائٹ سائنڈ انٹیجر میں ذخیرہ کی جاسکتی ہے جو 127_{10} ہے۔ جیسا کہ ایک بٹ کو ذخیرہ کرنے کے لیے دستیاب بٹس قیمت $n-1$ ہے۔ لہذا زیادہ سے زیادہ قیمت $2^{n-1} - 1$ ہوگی۔ ہم اس فارمولے کو 2 اور 4 بائٹس کے لیے زیادہ سے زیادہ اقدار کا حساب لگانے کے لیے استعمال کر سکتے ہیں۔ منفی اقدار کو مندرجہ ذیل سیکشن میں دیے گئے 2 کے کمپلیمنٹ کی صورت میں محفوظ کیا جاتا ہے۔

2.2.2.3 منفی اقدار اور 2 کے کمپلیمنٹ

منفی نمبروں کو ذخیرہ کرنے کے لیے کمپیوٹر ایک طریقہ استعمال کرتے ہیں جسے 2 کا کمپلیمنٹ کہا جاتا ہے۔ بائری نمبر 2 کا کمپلیمنٹ معلوم کرنے کے لیے ان مراحل پر عمل کریں:

(i) تمام بٹس کو تبدیل کریں (0 کو 1 سے اور 1 کو 0 سے تبدیل کریں)۔

(ii) لیسٹ سگنی فکینٹ بٹ (LSB) میں ایک جمع کریں۔

مثال: آئیے اعشاریہ نمبر 5 - کو 8 بٹ بائری نمبر میں تبدیل کریں:

(i) 00000101_2 : 5 کی بائری نمائندگی کے ساتھ شروع کریں۔

(ii) تمام بٹس کو الٹ دیں: 11111010_2

(iii) اب اس میں 1 جمع کریں: 11111010_2

+1

(iv) اس کا جواب آئے گا: 11111011_2

لہذا 8 بٹ میں 5 - کا دو کا کمپلی منٹ 11111011_2 ہے۔

چھوٹے سے چھوٹا انٹجر عدد

کسی 8-بٹ انٹجر کی قیمت کے لیے، ہم منفی قیمت کے لیے سائن بٹ اور تمام دوسری بٹس کو آن کر دیتے ہیں جس کا نتیجہ 11111111_2 ہوتا ہے۔ پہلی بٹ کے علاوہ، ہم تمام بٹس کا 2 کا کمپلی منٹ لیتے ہیں اور ہمیں 10000000_2 حاصل ہوتا ہے، جو 128_{10} ہے۔

لہذا، 1-بائٹ انٹجر میں کم از کم قیمت 128- ہوتی ہے، یعنی 2^{-7} اس لیے کم از کم قیمت کو 2^{n-1} فارمولے کے ذریعے معلوم کیا جاتا ہے،

جہاں n کل بٹس کی تعداد ہے۔

• 2-بائٹ انٹجر (16 بٹس): $2^{15} - 1 = -32,767$ کم از کم قیمت

• 4-بائٹ مکمل عدد (32 بٹس): $2^{31} - 1 = -2,147,483,647$ کم از کم قیمت

کیا آپ جانتے ہیں؟

کمپیوٹر میں بائری اور ریج استعمال کرنے کی وجہ یہ ہے کہ اس میں ٹرانزسٹر ہوتے ہیں۔

دلچسپ معلومات:

مختلف انٹجر کی اقسام کے ساتھ کام کرتے وقت، ہمیشہ چیک کریں کہ آیا ڈیٹا کی قسم غیر متوقع نتائج سے بچنے کے لیے سائنڈ ہے یا ان سائنڈ ہے، خاص طور پر جب بڑی قیمتوں یا منفی نمبروں سے نمٹنا ہو۔

یہ سمجھنا کہ مکمل اعداد کو میموری میں کس طرح ذخیرہ کیا جاتا ہے آپ کو کمپیوٹرز کے اندرونی کام کو سمجھنے میں مدد کرتا ہے اور اس بات کو یقینی بناتا ہے کہ آپ پروگرامنگ میں ڈیٹا کی مختلف اقسام کے ساتھ مؤثر طریقے سے کام کر سکتے ہیں۔

2.3 کمپیوٹر میموری میں حقیقی اقدار کو سنور کرنا

کمپیوٹرز میں، حقیقی اقدار جسے فلوٹنگ پوائنٹ نمبر بھی کہا جاتا ہے، کا اطلاق فریکشن اعشاریہ کے ساتھ نمبر کی نمائندگی کرنے کے لیے کیا جاتا ہے۔

2.3.1 فلوٹنگ پوائنٹ نمائندگی کو سمجھنا

سب سے پہلے، فلوٹنگ پوائنٹ نمبر (حقیقی اقدار) کو ذیل میں دیئے گئے سائنٹیفک نوٹیشن کی طرح پیش کیا جاتا ہے:

اس کا مطلب یہ ہے کہ: $2^{\text{exponent}} \times \text{مینٹسا} \times \text{سائن} = \text{فلوئنگ پوائنٹ نمبر}$

مندرجہ بالا فارمولے کے مطابق، $5.75 \times 2^{1.4375}$ کے طور پر پیش کیا جاتا ہے۔ ایک حقیقی (فلوئنگ پوائنٹ) قدر کے جزوی حصے کے لیے، اعشاریہ (بیس 10) سے بائری (بیس 2) میں تبدیل کرنے کا مطلب یہ ہے کہ فریکشنل حصے کو 2 کے ساتھ ضرب دی جائے اور نتیجے کا ایک صحیح حصہ لکھا جائے۔ یہ نئے جزوی حصے کے ساتھ کیا جاتا ہے جب تک کہ جزوی حصے کی قیمت صفر نہ ہو جائے یا ضرورت کے مطابق نہ ہو جائے۔

تبدیلی کے مراحل:

(i) فریکشنل پارٹ کی شناخت کریں:

اعشاریہ نمبر کا بقیہ حصہ حاصل کریں جو ایک فریکشن کی شکل میں ہے۔ مثال کے طور پر ایک نمبر 4.625 لیتے ہیں جس کا فریکشنل حصہ 625 ہے اور جزوی حصہ 4 ہے۔

(ii) فریکشنل پارٹ کو بائری میں تبدیل کریں:

نمبر کے آخر میں 0 شامل کریں اور 2 سے تقسیم کریں، پھر حاصل شدہ مکمل عدد لکھیں۔ اس عمل کو نئے حاصل ہونے والے حصے کے ساتھ اس وقت تک جاری رکھیں جب تک کہ یہ 0 تک نہ پہنچ جائے یا اس وقت تک جب تک کہ آپ کی مطلوبہ اعشاریہ جگہوں کی تعداد حاصل نہ ہو جائے۔

مثال: 0.375 کو بائری میں تبدیل کرنا۔

(i) فریکشنل حصے کی شناخت کریں۔

(ii) فریکشنل حصہ 0.375 کو بائری میں تبدیل کریں:

$$0.375 \times 2 = 0.75 \quad (\text{مکمل عدد حصہ } 0)$$

$$0.75 \times 2 = 1.5 \quad (\text{مکمل عدد حصہ } 1)$$

$$0.5 \times 2 = 1.0 \quad (\text{مکمل عدد حصہ } 1)$$

حاصل شدہ عددی حصے 0، 1، 1 ہیں۔

(iii) نتائج کو یکجا کریں: اوپر سے نیچے تک مکمل عدد کے حصوں کی بائری نمائندگی کو یکجا کریں:

$$0.375_{10} = 0.011_2$$

کمپیوٹنگ میں، حقیقی اعداد کو بائری شکل میں ظاہر کرنا ضروری ہے کیونکہ یہ پروسیسنگ کرنے اور ڈیٹا اسٹوریج میں سہولت فراہم کرتا ہے۔ اس عمل میں انٹجر (اعشاریہ) کے ساتھ ساتھ کسی مخصوص عدد کے صحیح حصے کو بائری میں تبدیل کرنا شامل ہے۔ جیسا کہ پہلے بھی بیان کیا گیا ہے کہ دو قسم کے سٹینڈرڈز بشمول ”سنگل پریسیژن (32 بٹ)“ اور ”ڈبل پریسیژن (64 بٹ)“ سب سے زیادہ استعمال ہوتے ہیں۔

2.3.1.1 سنگل پریسیژن (32 بٹ)

اس سٹینڈرڈ میں، 4 بائٹس یا 32 بٹس تفویض کیے جاتے ہیں جہاں پہلا بٹ ایک سائن بٹ

ہوتا ہے اور اگلے 8 بٹس ایکسپوننٹ بٹ کے ہوتے ہیں اور باقی 23 بٹس مینٹیسہ کے ہوتے ہیں۔
یہاں ایکسپوننٹ 126- اور 127+ کے درمیان کہیں بھی ہو سکتا ہے۔
قدروں کی تخمینہ حد: 1.4×10^{-45} سے 3.4×10^{38}

میںٹیسہ (23 بٹس)	ایکسپوننٹ (8 بٹس)	سائن بٹ	نمائندگی	قیمت
23 بٹس	8 بٹس	1 بٹ		گروپنگ
10111000000000000000000000000000	100000001	0	1.4375×2^2	5.75
10111000000000000000000000000000	100000001	1	1.4375×2^2	-5.75
01000000000000000000000000000000	011111101	0	1.25×2^{-3}	0.15625
01000000000000000000000000000000	011111101	1	1.25×2^{-3}	-0.15625

ٹیبل 2.3: 32 بٹ فلوٹنگ پوائنٹ نمائندگی

وضاحت:

ٹیبل 2.3 اس بات کی وضاحت کرتا ہے کہ بائری شکل میں 32 بٹ فلوٹنگ پوائنٹ کی قدروں کو کس طرح ظاہر کیا جاتا ہے۔ ہر فلوٹنگ پوائنٹ کی ویلیو کو تین اہم اجزاء میں تقسیم کیا جاتا ہے: سائن بٹ، ایکسپوننٹ اور مینٹیسہ۔
گروپنگ: یہ 32 بٹ فلوٹنگ پوائنٹ فارمیٹ کے لیے بٹ ایلوکیشن کی وضاحت کرتی ہے: سائن کے لیے 1 بٹ، ایکسپوننٹ کے لیے 8 بٹس اور مینٹیسہ کے لیے 23 بٹس۔

• 5.75 کی نمائندگی: 1.4375×2^2

سائن بٹ: 0 (مثبت)

ایکسپوننٹ: $2 + 127 = 129$

مینٹیسہ: 0.4375 کی بائری نمائندگی $10111000000000000000000000000000_2$ ہے۔

• -5.75 کی نمائندگی: -1.4375×2^2

سائن بٹ: 1 (منفی)

ایکسپوننٹ: $2 + 127 = 129$

مینٹیسہ: 0.4375 کی بائری نمائندگی $10111000000000000000000000000000_2$ ہے۔

• 0.15625 کی نمائندگی: -1.25×2^{-3}

سائن بٹ: 0 (مثبت)

ایکسپوننٹ: $-3 + 127 = 124$

مثلیسا: 0.25_2 کی بائری نمائندگی $010000000000000000000000_2$ ہے۔

• 0.15625 کی نمائندگی: -1.25×2^{-3}

سائن بٹ: 1 (منفی)

ایکسپوننٹ: $-3 + 127 = 124$

مثلیسا: 0.25 کی بائری نمائندگی $010000000000000000000000_2$ ہے۔

یہ بریک ڈاؤن اس بات کی وضاحت کرنے میں مدد کرتا ہے کہ کمپیوٹر سسٹم میں فلوٹنگ پوائنٹ اقدار کو کس طرح ذخیرہ اور ان کی پروسیسنگ کس طرح کی جاتی ہے۔

2.3.1.2 ڈبل پریزین (64 بٹ)

ڈبل پریزین میں، ایکسپوننٹ کو 11 بٹس کا استعمال کرتے ہوئے ظاہر کیا جاتا ہے۔ ایکسپوننٹ کو 1023 کے بائیس کے ساتھ بائیسڈ فارم میں ذخیرہ کیا جاتا ہے۔ اصل ایکسپوننٹ قدروں کی حد کا تعین مندرجہ ذیل طور پر کیا جاسکتا ہے:

بائیس (Bias): 1023

ایکسپوننٹ ریج:

ایکسپوننٹ ویلیوز 1022 - سے 1023 - تک ہوتی ہیں۔

لہذا ڈبل پریزین میں سب سے چھوٹی اور سب سے بڑی ممکنہ ایکسپوننٹ قدریں یہ ہیں:

کم از کم ایکسپوننٹ: -1022

زیادہ سے زیادہ ایکسپوننٹ: $+1023$

ہم مندرجہ بالا قیمتوں کو منفی کرنے کے علاوہ سنگل پریزین کے لیے بھی ہم یہی اقدامات کر سکتے ہیں۔

کیا آپ جانتے ہیں؟

چھوٹے سے چھوٹا مثبت عدد جو سنگل پریزین کا ذمہ دار ہوتا ہے 1.4×10^{45} ہے اور ڈبل پریزین میں 4.9×10^{-224} ہے۔

دلچسپ معلومات:

جب فلوٹنگ پوائنٹ قدروں کے ساتھ حساب کتاب کرتے ہیں تو ایک کو ممکنہ راؤنڈ آف کی غلطیوں پر بھی غور کرنا چاہئے۔ سائنٹفک کمپیوٹنگ میں، ان غلطیوں کو دیکھنا ضروری ہوتا ہے تاکہ درستگی کو برقرار رکھا جاسکے۔

سرگرمی

- درج ذیل اعشاریہ نمبروں کی بائری نمائندگی لکھیں: $10.5, 7.25, 2.5$
- پھر ان کے بائری فارمیٹ کو سنگل پریزین فارمیٹ میں تبدیل کریں۔

- اس آپریشن کو مکمل کرنے کے بعد اپنے ہم جماعت اور اپنے آپ سے پوچھیں کہ ایک نمبر کے سائز کی بنیاد پر نمائندگی کی درستی کی ڈگری کس طرح مختلف ہوتی ہے؟

کمپیوٹر میموری میں حقیقی اقدار کو کس طرح ذخیرہ کیا جاتا ہے اس کے بارے میں معلومات ہمیں ڈیجیٹل کمپیوٹیشن کی درستی اور حدود کو سمجھنے کے قابل بناتی ہے۔ فلوٹنگ پوائنٹ نمائندگی کی تفہیم کے ساتھ، ان نمبروں کو مختلف طریقوں سے کنٹرول کرنا اور استعمال کرنا ممکن ہے۔

2.4 بانسری اریٹھمٹک آپریشنز

اریٹھمٹک آپریشنز میں جمع، تفریق، ضرب تقسیم کے عوامل شامل ہیں جو ایک وقت میں دو نمبروں پر انجام دیے جاتے ہیں۔ بانسری اریٹھمٹک آپریشنز ڈیسمل کے آپریشنز سے ملتے جلتے ہیں لیکن بانسری کے قوانین پر عمل کرتے ہیں۔ یہاں بنیادی آپریشنز کا ایک مختصر جائزہ ہے:

2.4.1 جمع

بانسری جمع صرف دو ہندسوں کا استعمال کرتی ہے یعنی 0 اور 1۔ یہاں ہم سیکھیں گے کہ بانسری میں نمبروں کو کیسے جمع اور منفی نمبروں سے کیسے نمٹا جاتا ہے۔

بانسری میں جمع کے قواعد

بانسری جمع ان سادہ اصولوں کی پیروی کرتی ہے:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0$$

(ایک (1) کیری ہٹ کے ساتھ)

بانسری جمع کی مثال

مثال 1:

$$1101$$

$$+1011$$

$$11000$$

اس مثال میں:

$$(1 \text{ کیری}) \quad 1 + 1 = 0$$

$$(1 \text{ کیری}) \quad 0 + 1 + 1 = 0$$

$$(1 \text{ کیری}) \quad 1 + 0 + 1 = 0$$

$$(1 \text{ کیری}) \quad 1 + 1 + 1 = 1$$

2.4.2 تفریق

بائنری اریٹھمٹک میں 2 کے کمپلیمنٹ کو جمع کرتے ہوئے تفریق کا عمل انجام دیا جاتا ہے یا سب ٹریٹڈ کی قیمت کو مائنوڈ میں جمع کر کے انجام دیا جاسکتا ہے۔

مثال: بائنری میں 6 کو 9 سے تفریق کریں۔

$$1001_2 = 9_{10}$$

$$0110_2 = 6_{10}$$

مرحلہ 1: سب ٹریٹڈ کا 2 کا کمپلیمنٹ معلوم کریں۔

$$0110_2 \text{ کے بٹس کو الٹ دیں تو حاصل ہوگا } 1001_2$$

حاصل شدہ اٹنے نمبر میں 1 جمع کریں:

$$1001_2 + 1_2 = 1010_2 = -6_{10}$$

مرحلہ 2: مائنوڈ اور سب ٹریٹڈ کے 2 کے کمپلیمنٹ کو جمع کریں۔

$$1001_2 + 1_2 = 1010_2 = 10011_2$$

مرحلہ 3: 10011_2 میں کیری بٹ کو چھوڑ دیں۔

$$0011_2 = 3_{10}$$

$$9 - 6 = 3 \text{ اس طرح}$$

2.4.3 ضرب

بائنری نمبر میں 2 نمبر ہیں جو 0 اور 1 پر مشتمل ہوتے ہیں۔ بائنری نمبروں کو ضرب دینے کے لیے ڈیسمل نمبروں کو ضرب دینے سے ملتے جلتے اصولوں کی پیروی کرنا ہوتی ہے، لیکن آسان قواعد کے ساتھ۔ یہاں ہم سیکھیں گے کہ بائنری نمبروں کو کس طرح ضرب دی جاتی ہے۔

بائنری نمبروں کو ضرب کرنے کے مراحل

- بائنری نمبروں کو لکھیں، لیسٹ سگنیفیکینٹ بٹ (سب سے دائیں بٹ) کے حساب سے ترتیب دیں۔
- دوسرے نمبر کے ہر بٹ کو پہلے نمبر کے ہر بٹ سے ضرب کریں، جس طرح ڈیسمل میں ضرب کا طریق اختیار کیا جاتا ہے۔
- جزوی نتائج کو دوسری رو سے شروع کرتے ہوئے ہر نئی رو کے لیے بائیں طرف ایک جگہ آگے منتقل کریں۔
- حتمی نتائج حاصل کرنے کے لیے تمام جزوی نتائج کو جمع کریں۔

مثال

آئیے! دو بائنری نمبروں 101_2 اور 11_2 کو ضرب دیتے ہیں۔

$$\begin{array}{rcl}
 & & \times 11 \\
 101 & & \\
 (101_2 \times 1_2 \text{ یہ ہے}) & & \\
 101x & & (101_2 \times 1_2 \text{ یہ ہے اور شفٹ لفٹ ہے}) \\
 1111_2 & & (101_2 \times 11_2 = 1111_2 \text{ لہذا})
 \end{array}$$

کیا آپ جانتے ہیں؟

ایک سی بی یو پیچیدہ ہدایات اور پروگرام کو چلانے کے لیے ہر سیکنڈ میں لاکھوں بائنری نمبروں کو ضرب کرتا ہے۔

2.4.3 تقسیم

بائنری تقسیم اعشاریہ تقسیم کی طرح ہے لیکن اس میں صرف دو عدد، 0 اور 1 شامل ہیں۔ بائنری تقسیم میں اعشاریہ نظام کی تقسیم کی طرح موازنہ، تفریق اور منتقلی جیسے اقدامات شامل ہیں۔

بائنری تقسیم کے مراحل

- موازنہ کریں: موجودہ تقسیم ہونے والے نمبر کے ساتھ تقسیم کنندہ کا موازنہ کریں۔
- تخفیف: اگر تقسیم کنندہ تقسیم ہونے والے نمبر سے کم یا اس کے برابر ہو تو اس نمبر کے حصے سے تقسیم کنندہ کو تفریق کریں۔
- شفٹ: تقسیم ہونے والے نمبر کی بائنری کے اگلے ہندسے سے بقیہ حاصل کو نیچے منتقل کریں۔
- دہرائیں: اس عمل کو اس وقت تک دہرائیں جب تک کہ تقسیم ہونے والے نمبر سے کم کے تمام ہندسے استعمال نہ ہو جائیں۔

مثال

1100₂ کو 10₂ سے تقسیم کریں۔

$$\begin{array}{r}
 10 \overline{) 1100} \quad 110 \\
 \underline{-10} \\
 10 \\
 \underline{-10} \\
 0
 \end{array}$$

(مرحلہ 1: 10 کا موازنہ پہلے دو ہندسوں 11 سے کریں، 10 کو 11 سے تفریق کریں)

(مرحلہ 2: اگلے عدد 0 کو نیچے لائیں)

(مرحلہ 3: 10 کا 10 سے موازنہ کریں، 10 سے 10 کو تفریق کریں)

(مرحلہ 4: اگلے ہندسے 0 کو نیچے لائیں، مزید ہندسے باقی نہیں ہیں)

نتیجہ: $1100_2 / 10_2 = 110_2$

سرگرمی

بائنری میں تقسیم کی مشق کرنا

مقصد: دی گئی مثالوں کے ذریعے بائنری تقسیم پر عمل کرنا اور سمجھنا۔

ہدایات:

• تین سے چار طالب علموں کے گروپ بنائیں۔

• ہر گروپ مندرجہ ذیل بائری تقسیم کے مسائل کو حل کرے گا:

(i) $10101_2 - 10_2$

(ii) $11100_2 - 11_2$

(iii) $100110_2 - 101_2$

• اپنی تقسیم کے عمل کے ہر مرحلے کو واضح طور پر لکھیں۔

• کلاس کے سامنے اپنے حل پیش کریں، ہر مرحلے اور اس کے پیچھے استدلال کی وضاحت کریں۔

2.5 عام طور پر استعمال ہونے والی ٹیکسٹ انکوڈنگ اسکیمیں

ٹیکسٹ انکوڈنگ اسکیمیں مختلف زبانوں اور علامتوں کے حروف کو ایک ایسی شکل میں پیش کرنے کے لیے ضروری ہیں جسے کمپیوٹر سمجھ سکتے ہیں اور ان میں دی گئی ہدایات پر عمل کر سکتے ہیں۔ یہاں کمپیوٹرز میں استعمال ہونے والی کچھ ٹیکسٹ انکوڈنگ اسکیمیں ہیں:

ASCII 2.5.1

آسکی (ASCII)، امریکن سٹینڈرڈ فار انفارمیشن انٹرچینج کا مخفف ہے۔ یہ ایک کریٹر انکوڈنگ کا معیار ہے جو کمپیوٹر اور اسی طرح کے ٹیکسٹ استعمال کرنے والے آلات کے لیے اپنایا جاتا ہے۔ ہر حرف تہی، نمبر یا کسی بھی علامت کو 0 سے 127 کے درمیان ایک کوڈ نمبر دیا جاتا ہے جیسا کہ ٹیبل 2.4 میں دکھایا گیا ہے۔

آسکی (ASCII) مختلف کمپیوٹرز اور یو اینسز کو ٹیکسٹ معلومات کا قابل اعتماد طور پر تبادلہ کرنے کی اجازت دیتا ہے۔ آئیے! آسکی (ASCII) میں اپنے ملک Pakistan کا نام انکوڈ کرتے ہیں۔

• "P" کے لیے آسکی (ASCII) کوڈ 80 ہے۔

• "a" کے لیے آسکی (ASCII) کوڈ 97 ہے۔

• "k" کے لیے آسکی (ASCII) کوڈ 107 ہے۔

• "i" کے لیے آسکی (ASCII) کوڈ 105 ہے۔

• "s" کے لیے آسکی (ASCII) کوڈ 115 ہے۔

• "t" کے لیے آسکی (ASCII) کوڈ 116 ہے۔

• "a" کے لیے آسکی (ASCII) کوڈ 97 ہے۔

• "n" کے لیے آسکی (ASCII) کوڈ 110 ہے۔

سرگرمی

- اپنا نام لکھیں۔
- اپنے نام میں ہر حرف کے لیے آسکی (ASCII) کوڈ تلاش کریں۔ آپ اپنی مدد کے لیے آسکی (ASCII) ٹیبل استعمال کر سکتے ہیں۔
- ہر آسکی (ASCII) کوڈ کو بائری میں تبدیل کریں۔
- بائری میں اپنا نام لکھیں۔

Character	ASCII Code	Character	ASCII Code
SP (space)	32	!	33
"	34	#	35
\$	36	%	37
&	38	'	39
(	40	)	41
*	42	+	43
,	44	-	45
.	46	/	47
0	48	1	49
2	50	3	51
4	52	5	53
6	54	7	55
8	56	9	57
:	58	;	59
<	60	=	61
>	62	?	63
@	64	A	65
B	66	C	67
D	68	E	69
F	70	G	71

H	72	I	73
J	74	K	75
L	76	M	77
N	78	O	79
P	80	Q	81
R	82	S	83
T	84	U	85
V	86	W	87
X	88	Y	89
Z	90	[	91
\	92	]	93
^	94	-	95
?	96	a	97
b	98	c	99
d	100	e	101
f	102	g	103
h	104	i	105
j	106	k	107
l	108	m	109
n	110	o	111
p	112	q	113
r	114	s	115
t	116	u	117
v	118	w	119
x	120	y	121
z	122	{	123
	124	}	125
~	126	DEL	127

جدول: 2.4

2.5.1 توسیع شدہ آسکی (ASCII)

ایک معیاری آسکی (ASCII) نیبل میں 128 حروف شامل ہیں، یہ ایک آسکی کا توسیعی ورژن ہے جس میں 256 حروف شامل ہیں۔ یہ توسیع شدہ آسکی (ASCII) 8 بٹس استعمال کرتا ہے اور اس میں اضافی علامتیں، صوتی اور دیگر حروف شامل ہیں۔ تاہم، اصل 128 حروف سب سے زیادہ استعمال ہوتے ہیں اور کمپیوٹرز میں ٹیکسٹ کی نمائندگی کی بنیاد ہیں۔

2.5.2 یونیکوڈ (Unicode)

یونیکوڈ دنیا کے کسی بھی تحریری نظام میں استعمال ہونے والے تمام گرافک حروف کی نقشہ سازی کی ایک کوشش ہے۔ آسکی (ASCII) کے برعکس جو صرف 128 حروف کی نمائندگی کرنے کے قابل تھا، یونیکوڈ انکوڈنگ کی مختلف شکلوں جیسے UTF-8، UTF-16 اور UTF-32 کے ذریعے ایک ملین سے زیادہ حروف کی نمائندگی کر سکتا ہے۔

UTF یونیکوڈ ٹرانسفارمیشن فارمیٹ کا مخفف ہے۔

UTF-8 2.5.2.1

یہ ایک متغیر لمبائی کی حامل انکوڈنگ اسکیم ہے، جس کا مطلب ہے کہ یہ کسی حرف کی نمائندگی کرنے کے لیے مختلف تعداد میں بائٹس (1 سے 4 تک) استعمال کر سکتا ہے۔ UTF-8، آسکی (ASCII) کے ساتھ پسماندہ مطابقت رکھتا ہے۔ اس کا مطلب یہ ہے کہ UTF-8 بغیر کسی مسئلے کے پرانی آسکی (ASCII) انکوڈنگ اسکیم کو سمجھ اور استعمال کر سکتا ہے۔ لہذا اگر ہمارے پاس آسکی (ASCII) کوڈ میں لکھی گئی ٹیکسٹ فائل ہے تو، یہ UTF-8 کے ساتھ بالکل ٹھیک کام کرے گی۔ یہ پرانے اور نئے دونوں ٹیکسٹ پڑھنے کے قابل ہے۔

مثال:

یونیکوڈ میں حرف A کا کوڈ U+0041 ہے جو بائری فارمیٹ میں 01000001 ہے اور جو کہ 8 بٹس یا 1 بائٹ پر مشتمل ہے۔ آئیے دیکھتے ہیں کہ یو UTF-8 میں اردو حروف کی نمائندگی کس طرح کی جاتی ہے:

مثال کے طور پر:

اردو حرف ”ب“ کا یونیکوڈ U+0628 ہے، اس کی بائری شکل 1101100010101000 ہے جو یعنی یہ 2 بائٹس پر مشتمل ہے۔

UTF-16 2.5.2.2

UTF-16 ایک اور متغیر کردار انکوڈنگ میکانزم ہے، یہ 2 بائٹس یا 4 بائٹس فی کریکٹر استعمال کرتا ہے۔ UTF-8 کے برعکس، یہ آسکی (ASCII) کوڈ کے ساتھ مطابقت نہیں رکھتا ہے اس کا مطلب ہے کہ یہ آسکی (ASCII) کوڈ کا ترجمہ نہیں کر سکتا ہے۔

مثال کے طور پر:

انکوڈنگ میں حرف ”A“ بائری 01000001 00000000 جو اعشاری نظام میں 65 یا 2 بائٹس کے برابر ہے

مثال کے طور پر:

آسکی (ASCII) میں اردو حرف ”ب“ بائری میں 00000110 00101000 ہے جو 2 بائٹ میموری ہے۔

UTF-32 2.5.2.3

UTF-32 انکوڈنگ کا ایک طریقہ ہے جو 4 بائٹس استعمال کرتا ہے۔ تمام حروف کو فی کردار 4 بائٹس میں ہی ذخیرہ کیا جاتا ہے۔ یہ اسے بہت آسان بنا دیتا ہے لیکن اس کے ساتھ ہی جب جگہ کے استعمال کی بات آتی ہے تو یہ تھوڑا سا پیچیدہ محسوس ہو سکتا ہے۔

مثال کے طور پر:

حروف تہجی میں "A" کی بائٹری شکل 00000000 00000000 00000000 01000001 ہے اور یہ 4 بائٹس پر مشتمل ہے۔

2.6 کمپیوٹرز میں تصاویر، آڈیو اور ویڈیو ذخیرہ کرنا

کیا آپ نے کبھی سوچا ہے کہ آپ کی پسندیدہ تصاویر، گانے اور فلمیں آپ کے کمپیوٹر یا فون پر کیسے محفوظ ہوتی ہیں؟ آئیے! یہ سمجھنے کے لیے ڈیجیٹل اسٹوریج کی دلچسپ دنیا میں غوطہ لگاتے ہیں کہ کمپیوٹر ان مختلف قسم کی فائلوں کو منیج کرتے ہیں۔

کیا آپ جانتے ہیں؟

ڈیٹا کا سائز عام طور پر بائٹ اور اس کے ملٹیپل میں بیان کیا جاتا ہے۔

- 1 بائٹ (B) = 8 بٹس
- 1 گیگابائٹ (GB) = 1024 میگابائٹس
- 1 ایکسابائٹ (EB) = 1024 پیٹابائٹس
- 1 کلو بائٹ (KB) = 1024 بائٹس
- 1 ٹیرابائٹ (TB) = 1024 گیگابائٹس
- 1 زیٹابائٹ (ZB) = 1024 ایکسابائٹس
- 1 میگابائٹ (MB) = 1024 کلو بائٹس
- 1 پیٹابائٹ (PB) = 1024 ٹیرابائٹس
- 1 یوٹابائٹ (YB) = 1024 زیٹابائٹس

2.6.1 تصاویر ذخیرہ کرنا

تصاویر چھوٹے لقطوں سے بنی ہوتی ہیں جنہیں پکسلز کہا جاتا ہے۔ ہر پکسل کا ایک رنگ ہوتا ہے اور ان تمام پکسلز کا امتزاج مکمل تصویر بناتا ہے۔ کمپیوٹر ان رنگوں کی نمائندگی کرنے کے لیے نمبروں کا استعمال کرتے ہوئے تصاویر ذخیرہ کرتے ہیں۔

رنگ کی نمائندگی:

رنگین تصویر میں، ہر پکسل کے رنگ کو تین نمبروں سے پیش کیا جاسکتا ہے: سرخ، سبز اور نیلا (RGB)۔ ان میں سے ہر ایک نمبر عام طور پر 0 سے 255 تک ہوتا ہے۔ مثال کے طور پر RGB قیمتوں (255, 0, 0) کے ساتھ ایک پکسل چمک دار سرخ ہوگا۔

تصاویر فائل فارمیٹس:

مندرجہ ذیل تصاویر کے لیے عام طور پر استعمال ہونے والے امیج فارمیٹس ہیں، مثلاً JPEG

(Joint Photographic Expert Group) - یہ جگہ بچانے کے لیے تصویر کو کمپریس کرتا ہے لیکن کچھ معیار کھو سکتا ہے۔

PNG (Portable Network Graphic) : یہ شفافیت کی حمایت کرتا ہے اور ڈیٹا کھوئے بغیر اعلیٰ معیار کو برقرار رکھتا ہے۔
GIF (Graphic Interchange Format) : یہ چند رنگوں والی تصاویر اور انیمیشن کے لیے استعمال کیا جاتا ہے۔

سرگرمی

پکسل آرٹ بنائیں

- ایک سادہ تصویر (جیسا کہ مسکراتا ہوا چہرہ) بنانے کے لیے گراف کاغذ کا استعمال کریں
- ہر پکسل کو رنگ دیں اور استعمال ہونے والے ہر رنگ کے لیے RGB اقدار لکھیں۔
- کلاس کے ساتھ اپنے پکسل آرٹ اور RGB اقدار کا اشتراک کریں۔

2.6.2 آڈیو کو ذخیرہ کرنا

آڈیو فائلوں کو صوتی لہروں کو پکڑ کر اور انھیں ڈیجیٹل ڈیٹا میں تبدیل کر کے ذخیرہ کیا جاتا ہے۔ اس عمل میں سمپلنگ اور کوانٹائزیشن شامل ہے۔
سمپلنگ اور کوانٹائزیشن

سمپلنگ : اس میں باقاعدگی سے ایک خاص وقفے کے بعد سے آواز کی لہر کو ریکارڈ کرنا۔ فی سیکنڈ نمونوں کی تعداد کو سمپلنگ ریٹ کہا جاتا ہے۔ زیادہ نمونے لینے کی شرح کا مطلب بہتر معیار ہے۔

کوانٹائزیشن : اس سے مراد ہر نمونے کو ایک نمبر میں تبدیل کرنا ہے۔ فی نمونہ زیادہ بٹس کا مطلب زیادہ درست آواز ہے۔

آڈیو فائل کے فارمیٹس:

- **MP3 (MPEG Audio Layer-3)**: ایک عام فارمیٹ جو جگہ بچانے کے لیے آڈیو کو کمپریس کرتا ہے لیکن معیار کم ہو سکتا ہے۔
- **WAV (Waveform Audio File Format)**: یہ ایک نان کمپریسڈ فارمیٹ ہے جو اعلیٰ معیار کو برقرار رکھتا ہے۔
- **AAC (Advanced Audio Coding)**: موثر کمپریشن کے ساتھ اعلیٰ معیار کی آڈیو کے لیے بہت ساری اسٹریمنگ سروسز کے ذریعے استعمال کیا جاتا ہے۔

کیا آپ جانتے ہیں؟

MP3 فارمیٹ کی ایجاد نے موسیقی سننے کے انداز کو ڈرامائی انداز میں تبدیل کر دیا ہے۔ اس میں چھوٹا فائل سائز ہونے کی وجہ سے گانوں کو انٹرنیٹ پر آسانی سے شیئر کیا جاسکتا ہے۔

یہ وضاحت کرنے کے لیے کہ ان اکائیوں کو کس طرح استعمال کیا جائے گا، مندرجہ ذیل مثالیں دیکھیں:

- ایک تصویری فائل کا سائز 500 کلو بائٹ (KB) ہو سکتا ہے۔
- ایک میوزک فائل تقریباً 5 میگا بائٹس (MB) ہو سکتی ہے۔

- ایک مکمل ہائی ڈیفینیشن (HD) فلم تقریباً 2 گیگابائٹس (GB) ہو سکتی ہے۔
- ایک بڑی ہارڈ ڈرائیو میں 1 ٹیرابائٹس (TB) کی گنجائش ہو سکتی ہے۔

2.6.3 ویڈیوز خیرہ کرنا

ایک ویڈیو کو آڈیو کے ساتھ ساتھ ایک کے بعد ایک تیزی سے دکھائی جانے والی بہت سی تصاویر پر مشتمل ہوتی ہیں۔ ویڈیو میں ہر تصویر کو فریم کہا جاتا ہے۔

فریم اور فریم ریٹ:

فریم ریٹ: فی سیکنڈ دکھائے جانے والے فریموں کی تعداد کے فریم فی سیکنڈ (FPS) میں پیمائش کی جاتی ہے۔ عام فریم ریٹ 24 FPS (فلموں میں استعمال) اور 30 FPS (ٹی وی میں استعمال) ہیں۔ زیادہ فریم ریٹ کے نتیجے میں ویڈیوز زیادہ بہتر کوالٹی کی بنتی ہیں۔ ویڈیو فائل فارمیٹس:

MP4 (MPEG-4): یہ وسیع پیمانے پر استعمال ہونے والا فارمیٹ ہے جو معیار کو برقرار رکھتے ہوئے ویڈیو کو مؤثر طریقے سے کمپریس کرتا ہے۔

AVI (Audio Video Interleave): یہ پرانا فارمیٹ ہے جس کے نتیجے میں فائل کا سائز بڑا ہو سکتا ہے۔

MKV (Matroska Video): اعلیٰ معیار کی ویڈیو اور متعدد آڈیو ٹریکس ریسب ٹائٹلز کے لیے مناسب ہے۔

دلچسپ معلومات:

آن لائن ویڈیوز دیکھتے وقت، آپ کے انٹرنیٹ کی رفتار کی بنیاد پر معیار تبدیل ہو سکتے ہیں۔ اسٹریمنگ سروسز بفرنگ سے بچنے کے لیے ویڈیو کے معیار کو ایڈجسٹ کرتی ہیں۔

2.6.4 کمپیوٹر ان فائلوں کو کس طرح محفوظ کرتے ہیں؟

یہ تمام فائلیں (تصاویر، آڈیو اور ویڈیو) بائینری ڈیٹا کے طور پر ذخیرہ کی جاتی ہیں، جس کا مطلب ہے کہ وہ 0 اور 1 کے ترتیب کے ذریعے سے ان کی نمائندگی کرتے ہیں۔

اسٹوریج ڈیوائسز:

ہارڈ ڈسک ڈرائیو (HDD): یہ ڈیٹا پڑھنے/لکھنے کے لیے گھومنے والی ڈسک کا استعمال کرتی ہیں۔ ان میں اسٹوریج کی بڑی گنجائش ہوتی ہے۔

سولڈ اسٹیٹ ڈرائیو (SSD): یہ تیز تر رسائی کے اوقات اور بہتر کارکردگی کے لیے فلیش میموری کا استعمال کرتی ہے۔

کلاؤڈ اسٹوریج (Cloud Storage): یہ فائلوں کو ریموٹ سرورز پر اسٹور کرتا ہے جو انٹرنیٹ کے ذریعے قابل رسائی ہوتی ہیں، جس سے چمک اور بیک اپ کی سہولت ملتی ہے۔

کیا آپ جانتے ہیں؟

انٹرنیٹ کے ذریعے قابل رسائی ریموٹ سرورز پر فائلوں کو اسٹور کرتا ہے اور ان کا بیک اپ فراہم کرتا ہے۔ IBM نے 1956ء میں پہلی ہارڈ ڈرائیو بنائی اور اس کا وزن ایک ٹن سے زیادہ تھا اور یہ صرف 5,000,000 بائٹس ذخیرہ کر سکتا تھا، جو آج کے ایک اعلیٰ معیار کے گانے کو اسٹور کرنے کے لیے ناکافی ہے۔

یہ ہمیں اس بات کی تفہیم اور تعریف پر لاتا ہے کہ تصاویر، آڈیو اور ویڈیوز کو کمپیوٹرز میں کس طرح ذخیرہ کی جاتی ہیں جس سے ہم اپنے موجودہ ڈیجیٹل دور کے پیچھے بنیادی ٹیکنالوجی پر حیرت زدہ ہو جاتے ہیں۔ چاہے آپ تصاویر کی شوٹنگ کر رہے ہوں، دھنوں سے لطف اندوز ہو رہے ہوں، یا فلمیں دیکھ رہے ہوں، یہ سب اس وجہ سے ہے کہ کمپیوٹر کس طرح معلومات کا کوئی بھی کرتے ہیں۔

خلاصہ

- کمپیوٹنگ میں، نمبرنگ سسٹم اہم ہیں کیونکہ وہ معلومات کی نمائندگی، ذخیرہ اور پروسسنگ کے لیے بنیادی تشکیل فراہم ہیں۔
- اعشاریہ نمبر کا سسٹم ایک عددی نظام ہے جس کی اساس 10 ہے اور اس میں شامل 0 سے 9 تک کے ہندسے شامل ہیں جو عام طور پر ہماری حقیقی زندگی میں استعمال ہوتے ہیں۔
- بائری اساس 2 کا نمبر سسٹم ہے جو صرف دو ہندسوں 0 اور 1 پر مشتمل ہے۔ ہر ہندسے کو 2 کی طاقت سے ظاہر کیا جاتا ہے۔
- آکٹل نمبر سسٹم ایک اور عددی نظام ہے جس کی اساس 8 ہے۔ اس طرح، اس کے آٹھ ہندسے (0-7) ہیں، ہر ہندسہ 8 کی طاقت کی نمائندگی کرتا
- ہیکسا ڈسیمیل نمبر سسٹم ایک اور قسم کا نمبرنگ سسٹم ہے جس کی اساس 16 ہے جہاں 0-9 نمبر اور اس کے بعد A-F حروف تہجی استعمال کیے جاتے ہیں۔
- انٹجر سے مراد غیر منفی مکمل اعداد کا مجموعہ ہے۔ ان میں صفر اور تمام مثبت اعداد شامل ہیں۔
- کمپیوٹر منفی قیمتوں کو اسٹور کرنے کے لیے عام طور پر 2 کمپلیمنٹ کا طریقہ استعمال کرتا ہے۔
- کمپیوٹرز میں حقیقی قیمتیں جنہیں فلوٹنگ پوائنٹ نمبرز کا نام دیا جاتا ہے، ان نمبروں کی نمائندگی کرنے کے لیے استعمال کی جاتی ہیں جو جزوی یا اعشاریہ نقطہ رکھتے ہیں۔
- ارتھمیٹک آپریشنز کا مطلب ہے ایک مخصوص بیس میں نمبروں پر جمع، تفریق، ضرب یا تقسیم کرنا ہے۔
- آسکی (ASCII) دراصل امریکن کوڈ فار انفارمیشن انٹر چینج کا مخفف ہے۔ یہ ایک معیار ہے جو اس قسم کے ٹیکسٹ کو استعمال کرنے والے کمپیوٹرز اور دیگر آلات میں ٹیکسٹ کو کوڈ کرنے کے لیے استعمال کیا جاتا ہے۔
- آسکی (ASCII) کے معیاری جدول کے حوالے سے، کسی کو یہ سمجھنا ہوگا کہ 128 حروف کے ساتھ سادہ ورژن اور 256 حروف کے ساتھ ایک بہتر ورژن دونوں موجود ہیں۔

- یونیکوڈ ایک کریکٹر انکوڈنگ ہے جو دنیا کے تحریری نظام میں استعمال ہونے والے تمام حروف کے لیے کوڈ فراہم کرنے کی کوشش کرتا ہے۔
- UTF-16 ایک اور متغیر کردہ انکوڈنگ تکنیک ہے جس میں حروف کو یا تو دو یا چار بائٹس کے ساتھ ظاہر کیا جاتا ہے۔
- UTF-32 فکسڈ بلاک سائز کی انکوڈنگ کا طریقہ کار ہے، جس کا مطلب ہے کہ ہر بلاک کا سائز مستقل ہے۔ کسی استثنا کے بغیر، ہر ایک لفظ کو ہمیشہ اس لفظ کی مخصوص خصوصیات کے باوجود 4 بائٹس کا استعمال کرتے ہوئے ظاہر کیا جاتا ہے۔
- تصاویر چھوٹے پوائنٹس پر مشتمل ہوتی ہیں جنہیں پکسل کہا جاتا ہے۔ ان میں سے ہر ایک مخصوص رنگ کا ہوتا ہے اور اس طرح کے پکسلز کا مجموعہ ایک مکمل شکل یا تصویر بناتا ہے۔
- آڈیو فائلوں کو کمپیوٹر انٹرفیس کا استعمال کرتے ہوئے ریکارڈ کیا جاتا ہے تاکہ انہیں اسٹور کیا جاسکے اور پھر صوتی لہروں کو ڈیجیٹل شکل میں تبدیل کیا جاسکے۔ اس کے مراحل میں ڈیٹا سیمپلنگ اور کوآڈیشن شامل ہیں۔

مشق

1۔ درست جواب کا انتخاب کریں:

(i) آسکی (ASCII) کا کیا مطلب ہے؟

- (الف) امریکن سٹینڈرڈ کوڈ فار انفارمیشن انٹرچینج (ب) ایڈوانسڈ سٹینڈرڈ کوڈ فار انفارمیشن انٹرچینج
(ج) امریکن سٹینڈرڈ کمیونیکیشن انٹرچینج (د) ایڈوانسڈ سٹینڈرڈ کمیونیکیشن انٹرچینج

(ii) مندرجہ ذیل میں سے کون سا نمبر درست بائنری نمبر ہے؟

- (الف) 101102 (ب) 11011 (ج) 110.11 (د) 1101A

(iii) معیاری آسکی (ASCII) انکوڈنگ میں کتنے بائٹس استعمال ہوتے ہیں؟

- (الف) 7 بائٹس (ب) 8 بائٹس (ج) 16 بائٹس (د) 32 بائٹس

(iv) مندرجہ ذیل میں سے کون سا آسکی (ASCII) پر یونیکوڈ کا کلیدی فائدہ ہے؟

- (الف) یہ فی کردار کم بائٹس استعمال کرتا ہے
(ب) یہ بہت سی مختلف زبانوں کے حروف کی نمائندگی کر سکتا ہے
(ج) یہ بائنری کے ساتھ پیچھے سے مطابقت رکھتا ہے
(د) یہ انگریزی زبان کے لیے مخصوص ہے

(v) ایک ایچجر کو ذخیرہ کرنے کے لیے کتنے بائٹس استعمال ہوتے ہیں؟

- (الف) 1 بائٹ (ب) 2 بائٹس (ج) 4 بائٹس (د) 8 بائٹس

(vi) سائنڈ اور ان سائنڈ اعداد کے درمیان بنیادی فرق کیا ہے؟

(الف) ان سائنڈ اعداد منفی نہیں ہو سکتے

(ب) ان سائنڈ اعداد کی ایک بڑی رینج ہے

(ج) ان سائنڈ اعداد کو فلوٹنگ پوائنٹ فارمیٹ میں ذخیرہ کیا جاتا ہے

(د) سائنڈ عدد صرف مثبت نمبروں کے لیے استعمال ہوتے ہیں۔

(vii) سنگل پریسزن میں ایکسپوننٹ کے لیے کتنے بٹس استعمال کیے جاتے ہیں؟

(الف) 23 بٹس (ب) 8 بٹس (ج) 11 بٹس (د) 52 بٹس

(viii) سنگل پریسزن فلوٹنگ پوائنٹ نمبروں کے لیے قدروں کی تخمینہ رینج کیا ہے؟

(الف) 1.4×10^{-45} سے 3.4×10^{38} (ب) 3.4×10^{38} سے 1.4×10^{-45}
(ج) 4.9×10^{-324} سے 1.8×10^{308} (د) 1.8×10^{308} سے 4.9×10^{-324}

(ix) وہ چھوٹے نقطے جو تصویر بناتے ہیں، ان کو کہا جاتا ہے؟

(الف) پکسلز (ب) بٹ (ج) بائٹس (د) عقدہ

(x) RGB رنگ ماڈل میں، RGB کا کیا مطلب ہے؟

(الف) ریڈ، گرین، بلیو (ب) ریڈ، گرے، بلیک
(ج) ریڈ، گرین، براؤن (د) ریڈ، گرین، بلیک

2۔ مختصر سوالات

(i) آسکی (ASCII) انکوڈنگ اسکیم کا بنیادی مقصد کیا ہے؟

(ii) آسکی (ASCII) اور یونیکوڈ کے درمیان فرق کی وضاحت کریں۔

(iii) یونیکوڈ مختلف زبانوں کے حروف کو کس طرح سنبھالتا ہے؟

(iv) ایک ان سائنڈ انٹجر کے لیے قیمتوں کی رینج کیا ہے؟

(v) وضاحت کریں کہ بائری میں منفی عدد کی نمائندگی کس طرح کی جاتی ہے۔

(vi) ان سائنڈ انٹجر کو استعمال کرنے کا کیا فائدہ ہے؟

(vii) بٹس کی تعداد انٹجر کی قیمتوں کی حد کو کس طرح متاثر کرتی ہے؟

(viii) کمپیوٹنگ میں عام طور پر ان مقداروں کے مکمل اعداد کیوں استعمال کیے جاتے ہیں جو منفی نہیں ہو سکتے ہیں؟

(ix) سنگل پریسزن کے فلوٹنگ پوائنٹ نمبروں کی رینج کا حساب کیسے لگایا جاتا ہے؟

(x) کمپیوٹنگ میں فلوٹنگ پوائنٹ نمائندگی کی حدود کو سمجھنا کیوں ضروری ہے؟

3۔ تفصیلی سوالات

- (i) وضاحت کریں کہ یونیکوڈ کا استعمال کرتے ہوئے حروف کو کس طرح انکوڈ کیا جاتا ہے۔ مختلف زبانوں کے حروف اور ان کے متعلقہ یونیکوڈ کوڈ پوائنٹس کی مثالیں فراہم کریں۔
- (ii) تفصیل سے بیان کریں کہ کمپیوٹر میموری میں مکمل اعداد کو کس طرح ذخیرہ کیا جاتا ہے۔
- (iii) اعشاریہ عدد کو اس کی بائری نمائندگی میں تبدیل کرنے کے عمل کی وضاحت کریں اور اس کے برعکس مثبت اور منفی دونوں عدد کی مثالیں شامل کریں۔

(iv) مندرجہ ذیل بائری کو حل کریں۔

(الف) 101_2 کو 11_2 سے ضرب دیں۔ (ب) 1100 سے 10 کی تقسیم

(v) درج ذیل بائری نمبر جمع کریں:

(الف) $110 + 101$

(ب) $1011 + 1100$

(vi) درج ذیل نمبروں کو 4 بٹ بائری میں تبدیل کریں اور انھیں شامل کریں:

(ب) $-5 + 3$

(الف) $7 + (-4)$

(vii) مندرجہ ذیل کو حل کریں:

(ب) $1010_2 - 0011_2$

(الف) $1101_2 - 0100_2$

(د) $1110_2 - 100_2$

(ج) $1000_2 - 0110_2$