مختصر سوالات

1: کمپیوٹر کی وضاحت کریں۔

جواب: کمپیوٹر ایک ایسا آلہ ہے جو ڈیٹا کو ان پٹ کے طور پر لیتا ہے ڈیٹا کو پروسیس کرتا ہے اور آؤٹ پٹ دیتا ہے۔

2: printf فنکشن کی وضاحت کریں؟

جواب: printf سکرین پر آؤٹ پٹ دکھانے کے لیے سی پروگرامنگ لینگویج میں ایک بلٹ ان (built-in) فنکشن ہے۔ اس کا نام "پرنٹ فارمیٹڈ" (print formatted) سے نکلا ہے جو فارمیٹ شدہ آؤٹ پٹ کو سکرین پر پرنٹ کرنے کے لیے استعمال ہوتا ہے۔ تمام ڈیٹا کی اقسام printf فنکشن کے ساتھ ظاہر کی جاسکتی ہیں۔

3: فارمیٹ سپیسفائر کیا ہے؟

جواب: فارمیٹ سپیسفائر ڈیٹا ٹائپ فیلڈ کی چوڑائی اور سکرین پر ظاہر ہونے والے متغیر کی ویلیو کی شکل کی نمائندگی کرتا ہے۔ ایک فارمیٹ سپیسفائر ہمیشہ علامت % سے شروع ہوتا ہے۔ فارمیٹ سپیسفائر ان پٹ اور آؤٹ پٹ دونوں سٹیٹمنٹ کے لیے استعمال ہوتا ہے۔

4: C لینگویج میں مختلف ڈیٹا کی اقسام میں استعمال ہونے والے فارمیٹ سپیسفائر لکھیں؟

جواب: مندرجہ ذیل ٹیبل سی لینگویج میں مختلف ڈیٹا ٹائپس کے مقابلے میں فارمیٹ سپیفائر دکھاتا ہے۔

ڈیٹا ٹائپ	فارمیٹ سپیسفائر
int	% d or % i
float	% f
char	% c

5: اگر int age = 35; , float height= 5.8; ہو تو اس سٹیٹمنٹ کی آؤٹ پٹ کیا ہوگی؟

جواب:

```
printf("My age is %d and my height is %. 2f", age, height);
 My age is 35 and my height is 5.80
```

6: اس کوڈ کی آؤٹ پٹ کیا ہوگی؟

```
#include<stdio.h>
void main()
{
printf("Sum of 23 and 45 is %d", 23 + 45);
}
```

جواب: Sum of 23 and 45 is 68

7: () scanf کا مقصد کیا ہے؟

جواب: scanf سی لینگوئج میں ایک بلٹ ان فنکشن ہے جو صارف سے ویری ایبلز میں ان پٹ لیتا ہے۔ scanf فنکشن میں ہم فارمیٹ سپیسفائر کی مدد سے ان پٹ ڈیٹا کی متوقع قسم کی وضاحت کرتے ہیں۔ اگر صارف انٹیجر ڈیٹا ٹائپ میں ڈیٹا دیتا ہے تو scanf میں بیان کردہ فارمیٹ سپیسفائر d % یا i % ہونا چاہیے۔

8: scanf فنکشن میں عام غلطی کیا ہے؟

جواب: scanf فنکشن میں & اوپریٹر کو بھول جانا بہت عام سی غلطی ہے۔ & اوپریٹر کے بغیر پروگرام ایگزیکیوٹ تو جاتا ہے مگر متوقع نتائج نہیں دیتا ہے۔

9: ()getch کا کیا استعمال ہے؟

جواب: () getch فنکشن صارف سے ایک کریکٹر لینے کے لیے استعمال کیا جاتا ہے۔ صارف کا داخل کردہ کریکٹر سکرین پر ظاہر نہیں ہوتا ہے۔ یہ فنکشن عام طور پر پروگرام کی ایگزیکیوشن کو روکنے کے لیے استعمال کیا جاتا ہے کیونکہ پروگرام اس وقت تک آگے نہیں چلتا جب تک کہ صارف کوئی بٹن نہ دبائے۔

10: کس لائبریری فنکشن کو ()getch شامل کرنے کے لیے استعمال کیا جاتا ہے؟

جواب: ()getch فنکشن کو استعمال کرنے کے لیے ہمیں لائبریری conio.h کو پروگرام کے ہیڈر سیکشن میں شامل کرنے کی ضرورت پڑتی ہے۔

11: سٹیٹمنٹ ٹرمینیٹر کی تعریف کریں؟

جواب: سٹیٹمنٹ ٹرمنیٹر کمپائلر کے لیے شناخت کنندہ ہے جو لائن کے اختتام کی نشاندہی کرتا ہے۔ سی لینگوئج میں سیمی کولن(;) بطور سٹیٹمنٹ ٹرمینیٹر استعمال ہوتا ہے۔ اگر ہم ہر سٹیٹمنٹ کے آخر میں سیمی کولن نہ لگائیں تو ایرر (Error) آجاتا ہے۔

12: سی لینگوئج میں اسکیپ سیکوئنس کا کیا مقصد ہے؟

جواب: اسکیپ سیکوئنس printf فنکشن میں واوین("") کے اندر استعمال ہوتے ہیں۔ جو printf کو استعمال کرتے ہوئے آؤٹ پٹ دکھانے کے اپنے طریقہ کار کو تبدیل کرنے پر مجبور کرتے ہیں۔

13: سی لینگوئج اسکیپ سیکوئنس کو مثال سے بیان کریں۔

جواب:

```
printf ("My name is \"Ali\"");
```

مندرجہ بالا سٹیٹمنٹ کی آؤٹ پٹ یہ ہے۔

My name is "Ali"

مندرجہ بالا مثال میں "\" ایک اسکیپ سیکوئنس ہے۔ یہ printf کو " کمپیوٹر اسکرین پر " ظاہر کرنے کا سبب بنتا ہے۔

14: اسکیپ سیکوئنس کی تشکیل بیان کریں؟

جواب: اسکیپ سیکوئنس دو حروف پر مشتمل ہے۔ پہلا کریکٹر ہمیشہ بیک سلیش (back slash) ("\") ہوتا ہے اور دوسرا کریکٹر مطلوبہ فنکشنیلٹی کے مطابق آتا ہے جسے ہم حاصل کرنا چاہتے ہیں۔ بیک سلیش (\ ) کو اسکیپ کریکٹر کہا جاتا ہے جو اسکیپ کے بارے میں نشاندہی کرنے کے لیے ہر اسکیپ سیکوئنس سے منسلک ہوتا ہے۔ اسکیپ کریکٹر اور اس کے بعد آنے والا کریکٹر سکرین پر ظاہر نہیں ہوتے ہیں لیکن یہ مخصوص کام کرتے ہیں جو ان سے منسوب کیا گیا ہے۔

15: n\ کا مقصد کیا ہے؟

جواب: اسکیپ کریکٹر کے بعد n واضح کرتا ہے کہ کرسر(cursor) کو اگلی لائن (New Line) کے شروع پر لے کر جاتا ہے۔ یہ اسکیپ سیکوئنس ایک سے زیادہ لائنوں پر آؤٹ پٹ پرنٹ کرنے کے لیے استعمال ہوتا ہے۔

16: ٹیب (t\) کا مقصد بیان کریں؟

جواب: اسکیپ سیکوئنس t\ ، I/O فنکشن کو بتاتا ہے کہ افقی طور پر اگلے ٹیب سٹاپ (Tab Stop) پر جانا ہے۔ ایک ٹیب اسٹاپ 8 سپیسز (spaces) کا مجموعہ ہوتا ہے۔ t\ کا استعمال کرسر کو اگلے ٹیب اسٹاپ پر لے جاتا ہے۔

17: آپریٹر کی تعریف کریں؟

جواب: آپریٹرز ریاضی کی علامتیں ہیں جو اوپرینڈ (operand) پر کچھ آپریشن کرتی ہیں۔ اوپرینڈز ویری ایبلز یا ویلیوز ہو سکتی ہیں۔

18: بنیادی آپریٹر کے نام لکھیں؟

جواب: C میں بہت سے آپریٹرز استعمال ہوتے ہیں ان میں سے کچھ یہ ہیں:

- اسائنمنٹ آپریٹر (Assignment operator)
- منطقی آپریٹرز (Logical operators)
- ارتھمیٹک آپریٹرز (Arithmetic operators)
- ریلیشنل آپریٹرز (Relational operators)

19: اسائنمنٹ آپریٹر کیا ہے؟

جواب: اسائنمنٹ آپریٹر ایک متغیر کو قیمت تفویض کرنے کے لیے استعمال کیا جاتا ہے یا ایک متغیر کی قیمت کو دوسرے متغیر میں رکھنے کے لیے استعمال ہوتا ہے۔ برابری کی علامت (=) کو سی لینگویج میں اسائنمنٹ آپریٹر کے طور پر استعمال کیا جاتا ہے۔ مثال:

```
int sum = 5;
```

20: ایک پروگرام لکھیں جو دو انٹیجر ویری ایبلز کی قیمتوں کا آپس میں تبادلہ (swap) کرے۔

```
#include<stdio.h>
void main()
{
        int a = 2, b = 3, temp;
        temp = a;
        a = b;
        b = temp;
        printf("Value of b after swapping: %d\n:, b);
        printf("Value of a after swapping: %d\n", a);
}
```

21: ارتھمیٹک آپریٹر کا استعمال کیا ہے؟

جواب: ارتھمیٹک آپریٹرز ڈیٹا پر ریاضی کے کام کرنے کے لیے استعمال ہوتے ہیں۔

22: تقسیم آپریٹر کیا ہے؟

جواب: تقسیم کا آپریٹر (/) بائیں اوپرینڈ کی قیمت کو دائیں اوپرینڈ کی قیمت پر تقسیم کرتا ہے۔ جیسے:

```
float result = 3.0 / 2.0;
```

سٹیٹمنٹ چلنے کے بعد متغیر کا نتیجہ 1.5 آتا ہے۔

23:ضرب آپریٹر کیا ہے؟

جواب: ضرب آپریٹر (*) ایک بائنری آپریٹر ہے جو دو نمبروں کو ضرب دیتا ہے۔ جیسے:

```
int multiply = 5 * 5;
```

اس سٹیٹمنٹ پر عمل درآمد کے بعد ویری ایبل (multiply) کی قیمت 25 ہو جائے گی۔

24:جمع کا آپریٹر کیا ہے؟

جواب: جمع کا آپریٹر (+) دو اوپرینڈ کو جمع کرتا ہے۔ جیسے:

```
int add = 10 + 10;
```

متغیر add کی قیمت 20 ہو جائے گی۔

25:سٹیٹمنٹ a = a + 1 اور a -- کو C لینگوئج میں کیوں استعمال کیا جاتا ہے؟

جواب: سٹیٹمنٹ ;a = a + 1 ویری ایبل a کی قیمت میں 1 کا اضافہ کرنے کے لیے استعمال ہوتی ہے۔ C لینگوئج میں اس سٹیٹمنٹ کو ;++a یا ;a++ بھی لکھا جاسکتا ہے۔اسی طرح ;--a یا ;a-- ویری ایبل a کی قیمت میں 1 کو کم کرنے کے لیے استعمال کیا جاتا ہے۔

26 : تفریق آپریٹر کیا ہے؟

جواب: تفریق کا آپریٹر (-) بائیں اوپرینڈ سے دائیں اوپرینڈ کو تفریق کرتا ہے۔ جیسے:

```
int result = 20 – 15;
```

تفریق کرنے کے بعد متغیر result کی قیمت 5 ہو جاتی ہے۔

27:ماڈولس آپریٹر کیا ہے؟

جواب: ماڈولس آپریٹر (٪) بائیں اوپرینڈ کو دائیں اوپرینڈ پر تقسیم کرتا ہے اور تقسیم کے بعد بچنے والی باقی رقم لوٹاتا ہے۔ماڈولس آپریٹر انٹیجر ڈیٹا ٹائپ پر کام کرتا ہے۔ جیسے:

```
int remaining = 14 % 3;
```

اگر ہم 14 کو 3 سے تقسیم کریں تو 2 باقی بچے گا۔اس لیے ویری ایبل remaining کی قیمت 2 ہو گی۔

28: C لینگوئج میں ارتھمیٹک سٹیٹمنٹ لکھتے ہوئے عام غلطی کیا کی جاتی ہے؟

جواب: اگرچہ C لینگوئج میں ارتھمیٹک سٹیٹمنٹس لکھتے ہوئے ایک عام غلطی الجبرا کے روز مرہ کے قوانین کو استعمال کرنا ہے۔ مثلاً 6 * y کو 6y لکھنا اور x * x * x کو x^3 وغیرہ۔اس کے نتیجے میں کمپائلر ایرر (Complier Error) آتا ہے۔

29:ریلیشنل آپریٹرز سی لینگوئج میں کیوں استعمال ہوتے ہیں؟

جواب: ریلیشنل آپریٹرز دو قیمتوں کے درمیان تعلق کا تعین کرنے کے لیے انکا موازنہ کرتے ہیں۔ ریلیشنل آپریٹرز شناخت کرتے ہیں کہ قیمتیں برابر ہیں یا برابر نہیں۔ایک قیمت دوسری قیمت سے بڑی ہے یا چھوٹی ہے۔ سی لینگوئج ہمیں نیومیرک اور char ٹائپ کے ڈیٹا پر ریلیشنل آپریٹرز انجام دینے کی اجازت دیتی ہے۔

30: اسائنمنٹ آپریٹر (=) اور برابر کے آپریٹر (==) میں کیا فرق ہے؟

جواب: سی لینگوئج میں (==) آپریٹر دو ایکسپریشن کی مساوات کی جانچ کے لیے استعمال ہوتا ہے کہ وہ برابر ہیں کہ نہیں۔ جبکہ (=) آپریٹر دائیں والی ایکسپریشن کو بائیں والے ویری ایبل سے منسوب کرتا ہے۔ڈبل ایکوئل (equal) آپریٹر (==) چیک کرتا ہے کہ دونوں طرف یہ آپریٹرز برابر ہیں یا نہیں۔ سنگل ایکوئل آپریٹر (=) دائیں اوپرینڈز کو بائیں اوپرینڈز سے منسوب کرتا ہے۔

31: منطقی آپریٹر اور ارتھمیٹک آپریٹر کے درمیان فرق بیان کریں؟

ارتھمیٹک آپریٹر	منطقی آپریٹر
ارتھمیٹک آپریٹرز ڈیٹا پر ریاضی کے کام کرنے کے لیے استعمال ہوتے ہیں۔ تقسیم، ضرب، جمع، تفریق اور ماڈولس ارتھمیٹک آپریٹر کی قسم ہیں۔	منطقی آپریٹرز بولین ایکسپریشن پر آپریشن سرانجام دیتے ہیں اور اس کے نتیجے میں جواب بھی بولین ایکسپریشن ہوتا ہے۔ ریلیشنل آپریشن کا جواب بولین ایکسپریشن ہے۔ ایک سے زیادہ ریلیشنل ایکسپریشن کا جائزہ لینے کے لیے منطقی آپریٹرز کو سرانجام دیا جاسکتا ہے۔

32. تقسیم آپریٹر اور ضرب آپریٹر میں کیا فرق ہے؟

ضرب آپریٹر	تقسیم آپریٹر
ضرب آپریٹر (*) ایک بائنری آپریٹر ہے جو دو نمبروں کو ضرب دیتا ہے۔ جیسے: int multiply = 5 * 5; اس سٹیٹمنٹ پر عمل درآمد کے بعد ویری ایبل (multiply) کی قیمت 25 ہو گی۔	تقسیم کا آپریٹر (/) بائیں اوپرینڈ کی قیمت کو دائیں اوپرینڈ کی قیمت پر تقسیم کرتا ہے۔ جیسے: float result = 3.0 / 2.0; سٹیٹمنٹ چلنے کے بعد متغیر کا نتیجہ 1.5 آتا ہے۔

33: منطقی آپریٹر کی وضاحت کریں؟

جواب: منطقی آپریٹرز بولین ایکسپریشن پر آپریشن سرانجام دیتے ہیں اور اس کے نتیجے میں جواب بھی بولین ایکسپریشن ہوتا ہے۔ ریلیشنل آپریشن کا جواب بولین ایکسپریشن ہے۔ ایک سے زیادہ ریلیشنل ایکسپریشن کا جائزہ لینے کے لیے منطقی آپریٹرز کو سرانجام دیا جاسکتا ہے۔

34: منطقی آپریٹر کا نام لکھیں؟

جواب: مندرجہ ذیل ٹیبل بنیادی منطقی آپریٹرز اور ان کی تفصیل کو ظاہر کرتا ہے۔

تفصیل	آپریٹر
Logical AND	&&
Logical OR	||
Logical NOT	!

35: AND آپریٹر (&&) کیا ہے؟اس کو ٹروتھ ٹیبل کی مدد سے بیان کریں؟

جواب: AND آپریٹر (&&) دو بولین ایکسپریشن کو بطور اوپرینڈ لیتا ہے اور اگر اس کے دونوں ایکسپریشن درست (true) ہوں تو نتیجہ درست آتا ہے۔ اگر کوئی بھی اایکسپریشن غلط (false) ہو تو جواب غلط آتا ہے۔ AND آپریٹر کے لیے ٹروتھ ٹیبل ذیل میں دکھایا گیا ہے۔

ایکسپریشن 1	آپریٹر	ایکسپریشن 2	رزلٹ
False	&&	False	**False**
False	&&	True	**False**
True	&&	False	**False**
True	&&	True	**True**

36: OR آپریٹر (||) کیا ہے؟

جواب: OR آپریٹر دو بولین ایکسپریشن لیتا ہے اور کم از کم ایک ایکسپریشن درست (true) ہونے پر جواب درست دیتا ہے۔

37: OR آپریٹر کے لیے ٹروتھ ٹیبل بنائیں؟

جواب: OR آپریٹر کے لیے ٹروتھ ٹیبل ذیل میں دکھایا گیا ہے:

<table>
<tr><th>ایکسپریشن 1</th><th>آپریٹر</th><th>ایکسپریشن 2</th><th>رزلٹ</th></tr>
<tr><td>False</td><td>||</td><td>False</td><td>False</td></tr>
<tr><td>False</td><td>||</td><td>True</td><td>True</td></tr>
<tr><td>True</td><td>||</td><td>False</td><td>True</td></tr>
<tr><td>True</td><td>||</td><td>True</td><td>True</td></tr>
</table>

38: NOT آپریٹر (!) کیا ہے؟ ٹروتھ ٹیبل بھی بنائیں؟

جواب: NOT آپریٹر بولین ایکسپریشن کی قیمت کی نفی یا الٹ کرتا ہے۔ یہ اسے درست (true) بنا دیتا ہے اگر یہ غلط ہے اور اگر یہ غلط (false) ہے تو اسے درست بنا دیتا ہے۔ آپریٹر کے لیے ٹروتھ ٹیبل ذیل میں دیا گیا ہے۔

ایکسپریشن	آپریٹر	رزلٹ
False	!	True
True	!	False

39: یونری اور بائنری آپریٹرز میں فرق کریں۔

یونری آپریٹر	بائنری آپریٹر
یونری آپریٹرز صرف ایک آپریشن پر لاگو ہوتے ہیں جیسے منطقی NOT(!) آپریٹر کے پاس صرف ایک اوپرینڈ ہوتا ہے۔ سائن آپریٹر (-) یونری آپریٹر کی ایک اور مثال ہے۔ جیسے 5- ۔	بائنری آپریٹرز کو آپریشن کرنے کے لیے دو ایکسپریشن کی ضرورت ہوتی ہے۔ مثلاً تمام ارتھمیٹک اور ریلیشنل آپریٹرز بائنری آپریٹرز ہیں۔ منطقی آپریٹرز && اور || بھی بائنری آپریٹرز ہیں۔

40: آپریٹرز کی ترجیح کیا ہے؟

جواب: اگر ایکسپریشن میں ایک سے زیادہ آپریٹرز ہیں تو سوال یہ پیدا ہوتا ہے کہ پہلے کسے حل کریں گے۔ اس مسئلے کو حل کرنے کے لیے ہر آپریٹر کو ترجیح دی گئی ہے۔ جس آپریٹر کی ترجیح زیادہ ہے وہ دوسرے آپریٹر سے پہلے حل ہوگا۔ اگر آپریٹر کی ایک جیسی ترجیح ہو تو دائیں طرف والے آپریٹر سے پہلے بائیں طرف والا آپریٹر حل کریں گے۔

41: ٹرنری آپریٹر کی وضاحت کریں؟

جواب: سی پروگرامنگ لینگویج میں ایک ٹرنری آپریٹر (Ternary Operator) ہے جو تین اوپرینڈز پر کام کرتا ہے۔

42: scanf() اور getch() میں فرق کریں؟

getch()	scanf()
getch () فنکشن صارف سے ایک کریکٹر لینے کے لیے استعمال کیا جاتا ہے۔ صارف کا داخل کردہ کریکٹر سکرین پر ظاہر نہیں ہوتا ہے۔ یہ فنکشن عام طور پر پروگرام کی ایگزیکیوشن کو روکنے کے لیے استعمال کیا جاتا ہے کیونکہ پروگرام اس وقت تک آگے نہیں چلتا جب تک کہ صارف کوئی بٹن نہ دبائے۔	scanf() سی لینگویج میں ایک بلٹ ان فنکشن ہے جو صارف سے ویری ایبلز میں ان پٹ لیتا ہے۔ scanf فنکشن میں ہم فارمیٹ سپیسفائر کی مدد سے ان پٹ ڈیٹا کی متوقع قسم کی وضاحت کرتے ہیں۔ اگر صارف انٹیجر ڈیٹا ٹائپ میں دیتا ہے تو scanf میں بیان کردہ فارمیٹ سپیسفائر %d یا %i ہونا چاہیے۔

43: مختلف آپریٹرز کی ترجیح بیان کریں؟

آپریٹر	ترجیح
( )	1
!	2
* , / , %	3
+ , -	4
> , < , >= , <=	5
== , !=	6
&&	7
||	8
=	9

taleemcity.com

44: clrscr () کا استعمال لکھیں؟

جواب: یہ فنکشن سی لینگویج ایڈیٹر کی آؤٹ پٹ سکرین کو صاف کرنے کے لیے استعمال ہوتا ہے۔

Syntax: clrscr ();