مختصر سوالات

س1: مسئلہ کا حل کیا ہے؟

جواب: مشکل اور پیچیدہ مسئلے کو حل کرنے کے عمل کو مسئلہ حل کرنا کہا جاتا ہے۔ کسی مسئلے کو حل کرنے کے لیے ایک منظم حکمت عملی اپنانی پڑتی ہے۔

س2: تقسیم اور فتح کی وضاحت کریں؟

جواب: تقسیم کرو اور فتح کرو کی حکمت عملی میں ایک پیچیدہ مسئلے کو چھوٹے چھوٹے مسائل میں تقسیم کیا جاتا ہے اور ان کا حل نکال کر ایک پیچیدہ مسئلے کا حل نکال لیا جاتا ہے۔ اس طرح ہمارے لیے ایک وقت میں ایک چھوٹے مسئلے پر توجہ مرکوز کرنا آسان ہو جاتا ہے بجائے اس کے کہ ہم ہر وقت پورے مسئلے کے بارے میں سوچیں۔

س3: فنکشن کی تعریف کریں؟

جواب: فنکشن سٹیٹمنٹس کا ایک بلاک ہے جو ایک خاص کام انجام دیتا ہے۔ ہر پروگرام میں ایک مین فنکشن ہوتا ہے جو یوزر کے کاموں کو سرانجام دیتا ہے۔ اسی طرح ہم دوسرے فنکشنز لکھ سکتے ہیں اور انہیں کئی بار استعمال کر سکتے ہیں۔

س4: فنکشن کی اقسام کے نام لکھیں؟

جواب: بنیادی طور پر فنکشن کی دو اقسام ہیں۔

1) بلٹ ان فنکشنز (Built-in Function) 2) یوزر ڈیفائنڈ فنکشنز (User-defined Function)

س5: بلٹ ان فنکشنز کی تعریف کریں؟

جواب: جن فنکشنز کی پہلے سے سی پروگرامنگ لینگویج میں وضاحت کی جاتی ہے انہیں بلٹ ان فنکشن کہا جاتا ہے۔ ان فنکشنز کو لائبریری فنکشن بھی کہا جاتا ہے۔ یہ فنکشنز لینگویج کے ایک حصے کے طور پر دستیاب ہیں۔ یہ فنکشنز ریڈی میڈ پروگرام ہیں۔

س6: یوزر ڈیفائنڈ فنکشنز کی تعریف کریں؟

جواب: وہ فنکشنز جو پروگرامر بناتا ہے انہیں یوزر ڈیفائنڈ فنکشنز کہتے ہیں۔ یہ فنکشنز یوزر کی ضرورت کے مطابق بنائے جاتے ہیں۔

س7: فنکشنز کے فوائد لکھیں؟

جواب: فنکشنز ہمیں درج ذیل فوائد فراہم کرتے ہیں۔

1. دوبارہ استعمال 2. کام کو الگ کرنا 3. پڑھنے کی اہلیت 4. مسئلہ کی پیچیدگی سے نمٹنا

س8: دوبارہ استعمال کی وضاحت کریں؟

جواب: فنکشنز کے ذریعے ہم کوڈ کو دوبارہ استعمال کر سکتے ہیں۔ اس کا مطلب یہ ہے کہ جب بھی ہمیں ایک فنکشن کی فنکشنلیٹی (functionality) کی ضرورت ہو ہم اس فنکشن کو کال کر سکتے ہیں۔ ہمیں بار بار سٹیٹمنٹس کا ایک ہی مجموعہ لکھنے کی ضرورت نہیں ہے۔

س9: کام کو الگ کرنا سے کیا مراد ہے؟

جواب: فنکشنز ہمیں ایک کام کے کوڈ کو دوسرے کاموں کے کوڈ سے الگ کرنے کی اجازت دیتے ہیں۔ اگر ہمیں ایک فنکشن میں کوئی مسئلہ درپیش ہے تو پھر ہمیں اس مسئلے کو دور کرنے کے لیے پورے پروگرام کو چیک کرنے کی ضرورت نہیں ہے۔ ہمیں صرف ایک فنکشن پر توجہ دینے کی ضرورت ہے۔

س10: پڑھنے کی اہلیت کی وضاحت کریں؟

جواب: پروگرام کو کئی فنکشنز میں تقسیم کرنا پروگرام کی پڑھنے کی اہلیت کو بہتر بناتا ہے۔

س11: مسئلہ کی پیچیدگی سے نمٹنا کی اصطلاح کی وضاحت کریں؟

جواب: اگر ہم پورے پروگرام کو ایک پروسیجر کے طور پر لکھیں تو پروگرام کی دیکھ بھال کرنا مشکل ہو جاتا ہے۔ فنکشنز پروگرام کو چھوٹے چھوٹے حصوں میں تقسیم کرتے ہیں اور اس طرح مسئلے کی پیچیدگی کم ہو جاتی ہے۔

س12: فنکشنز سگنیچر کا استعمال کیا ہے؟

جواب: فنکشن کے سگنیچر (signature) فنکشن کی ان پٹ اور آؤٹ پٹ کی وضاحت کے لیے استعمال ہوتے ہیں۔

س13: فنکشن کا پیرامیٹر کیا ہے؟

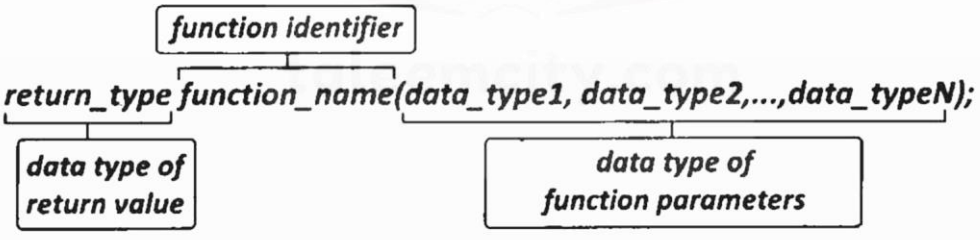
جواب: فنکشن سٹیٹمنٹس کا ایک بلاک ہے جو کچھ ان پٹ حاصل کرتا ہے اور کچھ آؤٹ پٹ فراہم کرتا ہے۔ فنکشن کی ان پٹس کو فنکشن کے پیرامیٹرز کہا جاتا ہے

س14: فنکشن کی ریٹرن ویلیو سے آپ کا کیا مطلب ہے؟

جواب: فنکشن سٹیٹمنٹس کا ایک بلاک ہے جو کچھ ان پٹ حاصل کرتا ہے اور کچھ آؤٹ پٹ فراہم کرتا ہے۔ فنکشن کی آؤٹ پٹ کو اس کی ریٹرن ویلیو کہا جاتا ہے۔ ایک فنکشن میں ایک سے زیادہ پیرامیٹرز ہو سکتے ہیں لیکن یہ ایک سے زیادہ قیمت ریٹرن نہیں کر سکتے۔

س15. فنکشن کے سگنیچر کا عمومی ڈھانچہ لکھیں۔

جواب:



س16: int square(int); فنکشن کیا کام کرتا ہے۔؟

جواب: ایک فنکشن جو ایک انٹیجر کو بطور ان پٹ لیتا ہے اور اس کا مربع ریٹرن کرے گا۔

س17: int float perimeter (float, float); فنکشن کیا کام کرتا ہے؟

جواب: ایک فنکشن جو مستطیل کی لمبائی اور چوڑائی کو بطور ان پٹ لیتا ہے اور مستطیل کا احاطہ ریٹرن کرے گا۔

س18: int largest (int, int, int); فنکشن کیا کام کرتا ہے؟

جواب: ایک فنکشن جو تین انٹیجر کو بطور ان پٹ لیتا ہے اور ان میں سب سے بڑی قیمت ریٹرن کرے گا۔

س19: float area (float); فنکشن کیا کام کرتا ہے؟

جواب: ایک فنکشن جو دائرے کے رداس کو بطور ان پٹ لیتا ہے اور دائرے کا رقبہ ریٹرن کرے گا۔

س20: int isVowel (char); فنکشن کیا کام کرتا ہے؟

جواب: ایک فنکشن جو ایک کریکٹر کو بطور ان پٹ لیتا ہے اور اگر حرف علت (vowel) ہے تو 1 ریٹرن کرے گا ورنہ 0 ریٹرن کرے گا۔

س21: فنکشن کو ڈیفائن کرنے کا عمومی ڈھانچہ بیان کریں؟

جواب: ایک فنکشن کے سگنیچر سے یہ نہیں پتہ چلتا کہ وہ کس کام کے لیے ہے۔ اس کے لیے فنکشن ڈیفینیشن ہوتی ہے۔ ایک فنکشن ڈیفینیشن کا ڈھانچہ کچھ اس طرح سے ہے۔

```
return_typefunction_name (data_type_var1, date_type var2, .... ,data_type varN)
{
        Body of the function
}
```

س22: فنکشن ڈیفینیشن کی وضاحت کے لیے مثال لکھیں۔

جواب: درج ذیل مثال ایک فنکشن showpangram( ) کی وضاحت کرتی ہے جو کوئی ان پٹ نہیں لیتا اور کچھ بھی ریٹرن نہیں کرتا بس کمپیوٹر سکرین پر " A Quick Brown Fox Jumps over the Lazy Dog " دکھاتا ہے۔

```
void showPangram()        <-- function name
{
        printf("\n A quick brown fox jumps over the lazy dog. \n");
}
```

جیسا کہ مذکورہ فنکشن کچھ ریٹرن نہیں کرتا اس طرح فنکشن کی ریٹرن ٹائپ void ہے۔

س23: کیا فنکشن ایک سے زیادہ قیمتیں واپس کر سکتا ہے؟

جواب: نہیں ایک فنکشن ایک سے زیادہ قیمتیں واپس نہیں کر سکتا۔

س24: فنکشن کال کرنے کے لیے عمومی ڈھانچہ بیان کریں؟

جواب: ہمیں کسی فنکشن کو کال کرنے کی ضرورت ہے تاکہ وہ پروگرام کو سونپا گیا کام انجام دے۔ مندرجہ ذیل عمومی ڈھانچہ ہے جو فنکشن کال کرنے کے لیے استعمال ہوتا ہے۔

```
function_name(value1, value2, ..., valueN);
```

س25: آرگیومنٹس کی وضاحت کریں؟

جواب: فنکشن کو دی جانے والی قیمتیں آرگیومنٹس کہلاتی ہیں۔

س26: فنکشن کے آرگیومنٹس اور پیرامیٹرز میں فرق کریں۔

جواب: جو قیمتیں فنکشن کو پاس کی جاتی ہیں وہ آرگیومینٹس کہلاتی ہے جبکہ فنکشن ڈیفینیشن میں جن ویری ایبلز میں یہ قیمتیں جاتی ہیں انہیں فنکشن کے پیرامیٹرز کہا جاتا ہے۔

س27: پروگرام میں فنکشنز کی ترتیب کے لیے کس نکتہ کو ذہن میں رکھنا چاہیے؟

جواب: پروگرام میں فنکشنز کی ترتیب کے لیے درج ذیل نکات کو ذہن میں رکھنا چاہیے۔

1. اگر کال کیے گئے فنکشن کی ڈیفینیشن کال کرنے والے فنکشن کی ڈیفینیشن سے پہلے ظاہر ہوتی ہے تو فنکشن کے سگنیچر کی ضرورت نہیں ہے۔
2. اگر کال کیے گئے فنکشن کی ڈیفینیشن کال کرنے والے فنکشن کی ڈیفینیشن کے بعد ظاہر ہوتی ہے تو کال کیے گئے فنکشن کا سگنیچر اس کال کرنے والے فنکشن سے پہلے لکھنا ضروری ہے۔

س28: فنکشن میں کتنے آرگیومنٹس استعمال کیے جاسکتے ہیں؟

جواب: سی لینگویج میں آرگیومنٹس کی تعداد پر کوئی پابندی نہیں ہے لیکن 8 سے زیادہ آرگیومنٹس کو ترجیح نہیں دی جاتی۔

س29: مندرجہ ذیل پروگرام کی آوٹ پٹ کیا ہے؟

```
main()
{
int x=20;
int y=10;
swap(x,y);
printf("%d %d",y,x+2);
}
swap(int x, int y)
{
int temp;
temp =x;
x=y;
y=temp;
}
```

Output:
10 22 (duplicate copy of arguments are generated on calling a function)

س30: درج ذیل کوڈ کی آوٹ پٹ کیا ہوگی؟

```
output()
{
printf("%p",output);
}
```

جواب: کچھ ایڈریس پرنٹ کیے جائیں گے کیونکہ فنکشن کے نام صرف ایڈریسز ہیں۔ اسی طرح آوٹ پٹ () بھی ایک فنکشن ہے اور اس کا ایڈریس پرنٹ کیا جائے گا۔

س31: درج ذیل کوڈ کی آوٹ پٹ کیا ہو گی؟

```
main()
{
int i;
printf("%d",scanf("%d",&i)); // value 10 is given as input here
}
```

جواب: کامیابی سے پڑھی گئی اشیاء کی تعداد بطور scanf ریٹرن 1 ہے۔ تو پڑھی گئی اشیاء کی تعداد 1 ہے۔

س32: سی پروگرام کی آوٹ پٹ کیا ہو گی؟

```
#include<stdio.h>
int main()
{
function();
return  0;
}
void function()
{
printf("Function in C is awesome");
}
```

Output: Compilation error.

س33: سی پروگرام میں ایرر کیا ہو گا؟

```
void message ();
{
printf ("Hope you are fine :");
return 23;
```

ایرر: فنکشن ڈیفینیشن کے بعد سیمی کالن نہیں آئے گا۔

س34: سی پروگرام میں ایرر کیا ہو گا؟

```
int max (int a; int b)
{
        if (a > b)
                return a;
        return b;
}
```

ایرر: پیرامیٹرز کے درمیان سیمی کالن کی بجائے کومہ آئے گا۔