```c
#include "stdio.h"
int main()
{
    char arr[100];
    printf("%d", scanf("%s", arr));
    /* Suppose that input value given
        for above scanf is "GeeksQuiz" */
    return 1;
}
```

- 9
- 1
- 10
- 100

```c
#include<iostream>
using namespace std;

class Point {
public:
    Point() { cout << "Normal Constructor called\n"; }
    Point(const Point &t) { cout << "Copy constructor called\n"; }
};

int main()
{
   Point *t1, *t2;
   t1 = new Point();
   t2 = new Point(*t1);
   Point t3 = *t1;
   Point t4;
   t4 = t3;
   return 0;
}
```

- Normal Constructor called
  Normal Constructor called
  Normal Constructor called
  Copy Constructor called
  Copy Constructor called
  Normal Constructor called
  Copy Constructor called
- Normal Constructor called
  Copy Constructor called
  Copy Constructor called

Normal Constructor called
Copy Constructor called

- Normal Constructor called
  Copy Constructor called
  Copy Constructor called
  Normal Constructor called

```
#include<iostream>
using namespace std;
class Point {
public:
    Point() { cout << "Constructor called"; }
};

int main()
{
   Point t1, *t2;
   return 0;
}
```

- compiler error
- constructor called constructor called
- constructor called

Output of following program?

```
#include<iostream>
using namespace std;
class Point {
    Point() { cout << "Constructor called"; }
};

int main()
{
   Point t1;
   return 0;
}
```

- compiler error
- runtime error
- constructor called

What is the output of the following program?

```
#include <stdio.h>
int main()
{
```

```
    int i = 0;
    switch (i)
    {
        case '0': printf("Geeks");
                  break;
        case '1': printf("Quiz");
                  break;
        default: printf("GeeksQuiz");
    }
    return 0;
}
```

- Geeks
- Quiz
- GeeksQuiz
- Compile-time error

```
#include <stdio.h>

int main()
{
    int i = 1024;
    for (; i; i >>= 1)
        printf("GeeksQuiz");
    return 0;
}
```

- 10
- 11
- infinite
- the program will show compile time error

```
#include "stdio.h"
int main()
{
  int x, y = 5, z = 5;
  x = y == z;
  printf("%d", x);
  getchar();
  return 0;
}
```

- 0
- 1
- 5
- compiler error

```c
#include <stdio.h>
int main()
{
    int i = 1, 2, 3;
    printf("%d", i);
    return 0;
}
```

- 1
- 3
- garbage value
- compile time error

```c
#include <stdio.h>
int main()
{
    int i = (1, 2, 3);
    printf("%d", i);
    return 0;
}
```

- 1
- 3
- garbage value
- compile time error

```c
#include "stdio.h"
int main()
{
  int x, y = 5, z = 5;
  x = y == z;
  printf("%d", x);
  getchar();
  return 0;
}
```

- 0
- 1
- 5
- compiler error

Predict the output of the below program:

```c
#include <stdio.h>
int main()
```

```
{
    printf("%d", 1 << 2 + 3 << 4);
    return 0;
}
```

- 112
- 52
- 512
- 0

```
#include <stdio.h>
#include <stdlib.h>
int top=0;
int fun1()
{
    char a[]= {'a','b','c','(','d'};
    return a[top++];
}
int main()
{
    char b[10];
    char ch2;
    int i = 0;
    while (ch2 = fun1() != '(')
    {
        b[i++] = ch2;
    }
    printf("%s",b);
    return 0;
}
```

- abc(
- abc
- 3 special characters with ASCII value 1
- Empty String

```
#include <stdio.h>

int fun(int n)
{
    if (n == 4)
       return n;
    else return 2*fun(n+1);
}


int main()
{
```

```
    printf("%d ", fun(2));
    return 0;
}
```

- 4
- 8
- 16
- runtime error

Consider the following recursive function fun(x, y). What is the value of fun(4, 3)

```
int fun(int x, int y)
{
  if (x == 0)
    return y;
  return fun(x - 1,  x + y);
}

int main(void)
{
    printf("%d",fun(4,3));
    return 0;
}
```

- 13
- 12
- 9
- 10

What does the following function print for n = 25?

void fun(int n) { if (n == 0) return;

printf("%d", n%2); fun(n/2); }

- 11001
- 10011
- 11111
- 00000

```
#include <stdio.h>

int main()
{
    int arr[5];

    // Assume that base address of arr is 2000 and size of integer
        // is 32 bit
```

```
    arr++;
    printf("%u", arr);

    return 0;
}
```

- 2002
- 2004
- 2020
- lvalue required

```
#include <stdio.h>

int main()
{
    int arr[5];
    // Assume base address of arr is 2000 and size of integer is 32 bit
    printf("%u %u", arr + 1, &arr + 1);

    return 0;
}
```

- 2004 2020
- 2004 2004
- 2004 garbage value
- the program fails to compile because address-of operator cannot be used with array name

```
#include <bits/stdc++.h>
using namespace std;

class Test
{
    public:
        Test() { cout << "Constructor called"; }
};

int main()
{
    Test *t = (Test *) malloc(sizeof(Test));
    return 0;
}
```

- constructor called
- empty
- compilation error
- runtime error

What does the following function do for a given Linked List with first node as head?

```
void fun1(struct node* head)
{
  if(head == NULL)
    return;
  fun1(head->next);
  printf("%d  ", head->data);
}
```

- prints all nodes of linked lists
- prints all nodes of linked list in reverse order
- prints alternate nodes of linked list
- prints alternate nodes in reverse order