What will be the output?

```
#include <stdio.h>
int main()
{
    {
        int a = 70;
    }
    {
        printf("%d", a);
    }
  return 0;
}
```

a) 70
b) Garbage value
c) Compilation error
d) None

Answer: c

Which of the following statements are correct in C?

a) A function with the same name cannot have different signatures
b) A function with the same name cannot have different return types
c) A function with the same name cannot have a different number of parameters
d) All of the mentioned

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

Answer: d

What will the function return?

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

a) x * y where x and y are integers
b) x * y where x and y are non-negative integers
c) x + y where x and y are integers
d) x + y where x and y are non-negative integers

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

Answer: b

What is the output of the following C program?

```
#include <stdio.h>
void foo(), f();
int main()
{
  f();
  return 0;
}
void foo()
{
   printf("2 ");
}
void f()
{
   printf("1 ");
   foo();
}
```

a) Compiler error as foo() is not declared in main
b) 1 2
c) 2 1
d) Compile time error due to declaration of functions inside main

Answer: b

What will be the output?

```
#include <stdio.h>
int main()
{
   switch(printf("C"))
   {
      default:
      printf("Default");
      case 1: printf("Choice1");
      break;
      case 2: printf("Choice2");
      break;
   }
   return 0;
}
```

a) Choice1
b) CChoice1
c) DefaultChoice1
d) CChoice1Choice2

Answer: b

What will be the output of the C code?

```
#include <stdio.h>
int main()
{
        char x=0;
        for(x=0; x<=127; x++)
        {
                printf("%d ", x);
        }
        return 0;
}
```

a) Compilation error
b) 0, 1, 2 …….., 127
c) 0, 1, 2, …….., 127, -128, -127,…infinite loop
d) 1, 2, 3……..,127

Answer: c

What is the output of the following C program?

```
#include <stdio.h>
int fun(int n)
{
   int i, j, sum = 0;
   for(i = 1; i<=n; i++)
      for(j=i; j<=i; j++)
         sum = sum + j;
   return(sum);
}
int main()
{
   printf("%d", fun(10));
   return 0;
}
```

a) 55
b) 45
c) 66
d) 10

Answer: a

Consider the function

```
find(int x, int y)
{
        return((x<y) ? 0 : (x-y));
}
```

Let a and b be two non-negative integers. The call find(a, find(a, b)) can be used to find the

a) Maximum of a, b
b) Positive difference between a and b
c) Sum of a and b
d) Minimum of a and b

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

Answer: d

```
int fibonacci (int n)
   {
   switch (n)
   {
   default:
   return (fibonacci(n - 1) + fibonacci(n - 2));
   case 1:
   case 2:
   }
   return 1;
   }
```

The function above has a flaw that may result in a serious error during some invocations.
Which one of the following describes the deficiency illustrated above?

(a) For some values of n, the environment will almost certainly exhaust its stack space before the calculation completes.
(b) An error in the algorithm causes unbounded recursion for all values of n.
(c) A break statement should be inserted after each case. Fall-through is not desirable here.
(d) The fibonacci() function includes calls to itself. This is not directly supported by Standard C due to its unreliability.

a) Option (a)
b) Option (b)
c) Option (c)
d) Option (d)

Answer: a

What is the output of the C code given below

```
#include <stdio.h>
float func(float age[ ]);

int main()
{
   float result, age[] = { 23.4, 55, 22.6, 3, 40.5, 18 };
   result = func(age);
   printf("Result is=%0.2f", result);
return 0;
}

float func(float age[ ])
{
   int i;
   float result, sum = 0.0;
   for (i = 0; i < 6; ++i) {
      sum += age[i];
   }
   result = (sum / 6);
   return result;
}
```

Answer: 27.08