Quiz-7

Question 1

Time: 00:00:04

What will be the output of the following pseudo

```
integer a, b, c;

set a = 11, b = 12, c = 10;

if (b > 0)

  b++

else

  a++

  end if

for (each b from 0 to 5)

   a = a + 1

  end for

   print (a + c)
```

- ◉ 26 — 60.57%
- ○ 20 — 6.65%
- ○ 22 — 25.54%
- ✕ 24 — 7.24%

```
int main()

  {

      int a=11,b=12,c=10;

      if(b>0)

      {

         b++;

      }

      else

      {

          a++;

      }

      for (b=0;b<5;b++)

      {

          a=a+1;

      }

      printf("%d",a+c);

  }
```

the condition if(b>0), will get true, the compiler will come down to the for loop, the loop will run for 5 iterations and in each iteration the value of a will be incremented by 1. So, the loop will increment the value of a by 5, thus the value of a will become 16
finally a+c will be printed, that is 16+10 = 26

Question 2

Time: 00:00:01

What will be the output of the following C code

```
#include<stdio.h>

int main ()

{

  int a = 0, i = 0, b;

  for (i = 0; i < 5; i++)

    {

      a++;

      while(a+i<5)

        printf ("Hello world\n");



    }

  printf ("%d", a);

  return 0;

}
```

- print hello world for 5 times, and then print 6
- print hello world for 5 times, and then print 1
- Code will stuck in infinite loop
- 5

In the above code the while condition will never gets false, hence the code will stuck into infinite loop

Question 3

Time: 00:00:00

What will be the output of the following C code

```
#include<stdio.h>

void main()

{

    int k=4;

    int *const p =&k;

    int r = 3;

    p = &r;

    printf("%d", p);

}
```

- It will print address of r
- It will print address of k + address of r
- It will print address of k
- Compile time error

Here p is a constant. So, if we try to assign a new value to p in the line p = &r; then that will result a compile time error.

Question 4

Time: 00:00:01

What will be the output of the following code ?

```
#include<stdio.h>

void main ()

{

  int k = 4;

  while(k>10)

  k=k*k;

  k++;

  printf ("%d", k);

}
```

- 4096
- 256
- 16
- 5

the while loop will not be executed since the condition will get false, the compiler will come out of the loop to the increment statement, it will increase the value of k by 1 and will print the final value of k, that is 5

Question 5

Time: 00:00:01

What will be the returned value for the pseudo-code for p=22, q=127

```
fun(int p,int q)



   if (p == 1)

   return q;

 else if (p%2==0)

   return fun (p - 1 , q);

 else

   return 0;
```

- 9 — 4.77%
- 6 — 6.01% ✗
- 5 — 3.98%
- None of the mentioned above — 85.25% ✓

Let's understand it step by step
1st iteration - p=22 q=127, as p%2=0 here so, fun(21,127) will be called
2nd iteration - p=21 q=127 as p not equals to 1 or p%2!=0 so, else condition will be executed.
That means 0 will be returned

Question 6

Time: 00:00:00

What will be the output of following code?

```
#include <stdio.h>

int foo(int* a, int* b)

{

   int sum = *a + *b;

   *b = *a;

    return *a = sum - *b;

}

int main()

{

   int i = 0, j = 1, k = 2, l;

   l = i++ || foo(&j, &k);

   printf("%d %d %d %d", i, j, k, l);

   return 0;

}
```

- 1121
- 1221
- 1211
- 1222

The control in the logical OR goes to the second expression only if the first expression results in FALSE. The function foo() is called because i++ returns 0(post-increment) after incrementing the value of i to 1. The foo() function actually swaps the values of two variables and returns the value . So, values of variables j and k get exchanged and OR expression evaluates to be TRUE.1 2 1 1

Question 7

Time: 00:00:06

Predict the output

```
#include <stdio.h>

int i;

int main()

{

  if (i);

  else

     printf("else");

  return 0;

}
```

- else
- zero
- if
- error

Since i is defined globally, it is initialized with default value 0. The else block is executed as the expression within if evaluates to FALSE. An empty block is equivalent to a semi-colon(;). So the statements if **(i);** and**if (i) {}** are equivalent.

Question 8

Time: 00:00:14

What will be the output of the following algorithm?

```
Start

Declare a=0, I and b

for I =0 to 3

Increment a by 1

if I = 3 then

print hello

get out of the loop

End if

End for

print a
```

- 4
- 1
- hello4
- hello

Inside the loop with value of i from 0 to 4, it will increment the value of a, 4 times that will be equal to 1, 2, 3 and 4. but first inside the for loop hello will be printed when the value of I = 3
and at last a will be printed i.e. 4 so the output will be hello4

C code corresponding to given pseudocode

```
#include<stdio.h>

int main()

{

int a=0,i,b;

for (i=0;i<=3;i++)

{

   a=a+1;

   if (i==3)

{

   printf("Hello");

   break;

}

}

printf("%d",a);

}
```

Question 9

Time: 00:00:02

What will be the output of the following pseudo code ?

```
Input x =5,y = 7

 x=x+1;

  y=y+1;

  x=x+y;

if(x%2==0) print x else print y
```

- 14
- 5
- 8
- 10

```
#include<stdio.h>

int main ()

{

  int x = 5,y=7;

  x=x+1;

  y=y+1;

  x=x+y;

  if(x%2==0)

  printf("%d",x);

 else

  printf("%d",y);

}
```

After x=x+1 the value of x=5+1=6 and after y=y+1 the value of y is 7+1=8
Now x=x+y so, x will be 6+8=14
As x%2=0 hence the if condition is true So, x or 14 will be printed.

Question 10

Time: 00:00:01

What will be the output of the following pseudocode?

```
Declare variable x, y

Set x = 0 and y = 1

    while(x<5)

  print x

  x=x+y

  x++

  End of loop
```

- 024
- 023
- 124
- 038

```c
#include<stdio.h>

int main()

{

    int x=0,y=1;

    while(x<5)

    {

        printf("%d ",x);

        x=x+y;

        x++;

    }

}
```

In the first iteration x=0 that will be printed and then the value of x will be x=x+y=1 and x++ or 1+1=2
So, in the next iteration x will be printed which is 2
Similarly for the 3rd iteration x will be 4 and it will be printed
As, for the next iteration x=6 so, the loop condition will be false.
Hence 0 2 4 will be the output.

Pseudocode quiz 2-7

1) (i) Initial Values: a = 11, b = 12, c = ?

(ii) Since b = 12 > 0, b is incremented to 13. (b++)

(iii) The loop runs 5 times (0 to 4): (because counter starts from 0) in each iteration, a is incremented by 1:

a = 11 + 5 = 16 (a = a + i)

(iv) Final output is a + c = 16 + 10 = 26

output: 26

M T W T F S S

COMPASS

Date :

for loop runs from 0 to 05:

b become 13
a become 17 (11+6)
output is a+c = 17+18 = 35,,

2) (i) Initialize : a = 0, i = 0

(ii) for loop : Runs for i = 0 to 4 (5 iterations)

(iii) In each iteration:
a is incremented by 1 (a++)
while (a+i<5)

• First (i = 0):
a = 1
while (1+0)<5 → true
⇒ prints "Helloworld" infinitely
because neither a nor i change inside
while loop.

⇒ while loop condition remains true
forever even it exceeds it limit whereas
nor i is modified inside while loop.

output: So, code get stuck in an infinite Loop

3) output :- compile time error

Here, p is a constant
⇒ * const p = &k (k = 4);

COMPASS

M T W T F S S

Date:

Here p is assigned with address of k

where one set of value is assigned by is a constant type of p.

So: p = &x where x = 4

=> It throws back with a compile time error => (new value is not taken)

4) initialize : k = 4

So, (4 > 10) is False & continue until it meet True. So,

Compiler will come out of the loop & executes in next statements.

So, k++ => 4++ => 5, & stop at

print(c) => output: 5

5) Steps:-

initialize p = 22, q = 127

So, First Iteration

p = 22, 127.

if (22 == 1) is false goes to else.

(22 % 2 == 0) is True

fun(21, 127) calls the fun

tion for the value to be returned.

2nd iteration

p = 21, 127

if (21 == -1) is false goes to first else,

(21 % 21 == 0)

goes to next else:

returns 0.

So, 0 value is returned to the function fn(21, 127)

output:- none of the above mentioned

6) (i) initial values:-

i = 0, j = 1, k = 2, l is also initialized.

(ii) fn foo(&j, &k) execution:-

- Sum = *a + *b = 1 + 2 = 3
- *b - *a = 1 → k = 1
- *a = Sum - *b = 3 - 1 - 2 → j = 2

returns 2.

(iii) logical OR result:-

1 = 0 || 2 = 1

(iv) • i = 1 (due to its Final value)

j = 2 (updated in foo)

COMPASS

M T W T F S S

Date :

- K = 1 (updated in loop)
  J = 1 (result of logical or)

=) Print (1 2 1) =) output :- 121

7) (i) i is a global int variable, which is initialized to 0 by default

(ii) if (i): =) if (0) is false
So, the condition is false:
control flows to else block.
(Since it contains 0)

(iii) Output :-
it print ("else") in else block
as string output "else".

8) 1. initialization :-
- a = 0
- Loop runs for I = 0, 1, 2, 3.

2. Loop Execution :-

- Iteration 1 (I = 0):
  =) a = 0 + 1 = 1
  =) I ≠ 3 → skip if block
- Iteration 2 (I = 1):
  =) a = 1 + 1 = 2
  =) I ≠ 3 → skip if block
- Iteration 3 (I = 2):
  =) a = 2 + 1 = 3
  =) I ≠ 3 → skip if block

- Iteration 4 (I = 3):
  → a = 3 + 1 = 4
  → I = 3 → print "hello" and exit loop //

3. Output:- print a → 4
The String "hello" is printed during the loop & 4 afterward.

→ hello 4 → output

9) initialize x = 5, y = 7

→ x = x + 1 → 5 + 1 = 6 ✓
y = y + 1 → 7 + 1 = 8 ✓
→ x = x + y → 6 + 8 = 14 ✓

if (14 % 2) == 0 → True ✓
print(x) → print(14)
→ out = 14

10) initialize x = 0, y = 1

(i) 0 < 5:
print(0)
x = x + y → 0 + 1 = 1
x++ → 1++ → x = 2

(ii) 2 < 5:
Print(2)
x = x + y → 2 + 1 = 3
x++ → 3++ → x = 4

(iii) $4 < 5$:

print(4)

x = x+y ⇒ 4+1 = 5

x++ ⇒ 5++ ⇒ $\boxed{x = 6}$

(iv) $6 < 5$:

it comes out of loop.

∴ output : $\boxed{x = 0, 2, 4}$