



Daffodil International University

Faculty of Science & Information Technology
Department of Computer Science and Engineering

Quiz-02(72_K)

Course Code: CSE113, Course Title: Programming and Problem solving

Time: 30 minutes

Marks: 15

1. Find the output of the program.

6

```
i)
#include <stdio.h>
int main()
{
    int tracker = 10;
    int points = 0;
    do {
        if (tracker % 3 == 0)
        {
            points = points + tracker;
            printf("Checkpoint Alpha: %d\n", points);
        }
        else {
            points = points + 1;
        }
        tracker = tracker - 2;
    } while (tracker > 2);
    printf("Final Status -> Tracker: %d, Points: %d\n", tracker, points);
    return 0;
}
```

Handwritten notes:
 $10 \rightarrow 8 \rightarrow 6$
 $6 \rightarrow 4 \rightarrow 2$

```
ii)
#include <stdio.h>
int main()
{
    int code = 1;
    int modifier = 5;
    switch (code + (modifier / 2))
    {
        case 1:
            printf("Result: Blue\n");
            break;
        case 3:
            printf("Result: Green\n");
            break;
        case 4:
            printf("Result: Yellow\n");
            break;
        case 5:
            printf("Result: Red\n");
            break;
        default:
            printf("Result: Purple\n");
    }

    return 0;
}
```

Handwritten notes:
 $5/2 = 2 \text{ remainder } 1$
 $1 + 2 = 3$

2. Find the error of the program

- #include <stdio.h>
- int main()
- {
- int counter = 5;
- int system_score = 100;
- do {
- switch (counter % 2)
- {
- case 1:
- system_score -= (counter * 3);

```

k.         printf("Mode Alpha - Score: %d\n", system_score);
l.         break;
m.         case 0;
n.         system_score += (counter * 2)
o.         printf("Mode Beta - Score: %d\n", system_score);
p.         break;
q.         default:
r.         printf("System stable.\n");
s.         } -5
t.         if (system_score < 70) { -1
u.         system_score = 70;
v.         printf("Warning: Low score override applied.\n");
w.         counter--;
x.         } -1 while (counter > 0)
y.         printf("Final Telemetry Score: %d\n", system_score);
z.         return 0;
aa.        } -1

```

3. In the mystical land of Numeria, two wizard brothers, Wizard A and Wizard B, guard the checkpoints of a sacred mountain path. Travelers are only allowed to pass through the path if they can find the secret "Harmony Numbers" hidden between the two checkpoint boundaries.

A number achieves harmony only if it is simultaneously blessed by the Spirits of 5 and the Spirits of 7 (meaning it must be perfectly divisible by both 5 and 7, or in other words, divisible by 35!).

Let's write a program to help the wizards automatically find these magical numbers between any two checkpoints a traveler chooses!

Input:

10 80

Output:

35 70

4. In the digital world of Neon-Tokyo, a dangerous virus has broken out in the mainframe. To protect the core database, the Chief Security Officer decides to construct a dynamic cryptographic firewall shield. The shield is made up of a grid of heavy data blocks represented by the hash symbol (#). To deploy the firewall correctly, the computer system must build a perfect square grid based on the safety level requested by the security team. The code generator writes the firewall row by row, placing one data block at a time from left to right.

Input:

5

Output:

```

#####
#####
#####
#####
#####

```

10-80
A B

for i=1; i<=n; i++
j=1; j<=n; j++
printf("#");

x