

Quiz 2: Set B Solution

Introduction to Computing and Programming (CSD101)

Max. Marks: 15

Date: 24-10-2024

Duration: 30 min.

Name: _____ Roll No. _____

Note: Provide reasoning to all the MCQ type questions

1. What is the output of the following Code snippet (1 mark)

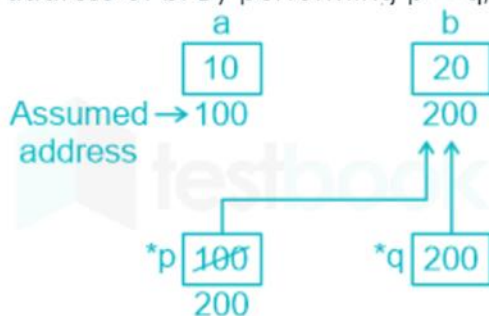
```
#include<stdio.h>
```

Solution:

```
int main()
{
    int a = 10, b = 20;
    int *p = &a, *q = &b;
    p=q;
    return 0;
}
```

- a. Both a and b will contain 20
- b. Both p and q will point to b
- c. Both p and q will point to a
- d. Both a and b will contain 10
- e. None of these

a and b are two integer variables. p stores address of a and q stores address of b. By performing p = q, p now stores the address of b.



2. What is the output of the following program? (2 marks)

```
#include <stdio.h>
```

Solution:

```
int main() {
    char c[] = "ICRBCSIT17";
    char *p = c;
    printf("%s\n", c+p[2]-p[6]-1);
}
```

Output:

17

c is base address, ascii value of p[2] = 83, ascii value of p[6] = 73

base address+83-73-1

base address + 9

so it will print all the character after 9th position

3.

A. The following statement

(1 mark)

Solution:

```
if (a>b)
if (c>b)
printf("One");
else
if (c==a) printf("Two");
else printf("Three");
else printf("Four");
a. Results in a syntax error
b. Prints Four in c<=b
c. Prints Two if c<=b
d. Prints Four in a<=b
```

Solution:

For a = 10; b = 20; c= 30; Four will be printed.

For a = 30; b = 20; c= 30; One will be printed

For a = 30; b = 20; c= 10; Three will be printed

B. The above statement can never be printed

(1 mark)

a. One

Solution:

b. Two

c. Three

d. Four

Solution:

Two will only execute if a>b and b>=c so a will not be equal to c

4. Choose the statements that are syntactically correct

(1 mark)

a. /* Is /* this is valid */ comment*/ **Solution:**

b. for(; ;);

c. return;

d. `return(5+2);`

Solution: Nesting cannot be applied to multiline comment statements.

5. What is Recursion, represent it with an example? Write two differences between Recursion and Iteration. (4 marks)

Recursion is a programming technique where a function calls itself to solve a smaller instance of the same problem. It typically consists of a base case, which stops the recursion, and a recursive case, which continues the recursion.

Example of Recursion

A classic example of recursion is the calculation of the factorial of a number.

```
int factorial(int n)

{ if (n == 0) // Base case

    return 1;

else return n * factorial(n - 1); // Recursive case

}
```

| Iteration | Recursion |
|--|---|
| Iteration explicitly user a repetition structure. | Recursion achieves repetition through repeated function calls. |
| Iteration terminates when the loop continuation. | Recursion terminates when a base case is recognized. |
| Iteration keeps modifying the counter until the loop continuation condition fails. | Recursion keeps producing simple versions of the original problem until the base case is reached. |
| Iteration normally occurs within a loop so the extra memory assigned is omitted. | Recursion causes another copy of the function and hence a considerable memory space's occupied. |
| It reduces the processor's operating time. | It increases the processor's operating time. |

6. Define two functions in C (**Note:** Do not write the whole program):
[a] A recursive function to count the number of digits for a given integer. (2.5 marks)

```
int countDigits(int n) {
    // Handle negative numbers by taking the absolute value
    if (n < 0) {
```

```

        n = -n; // Convert to positive
    }

    // Recursive case: strip off the last digit and count
    if (n < 10) {
        return 1; // If now n is a single digit, return 1
    } else {
        return 1 + countDigits(n / 10); // Count the last
digit and continue
    }
}

```

If the return type is different and they have printed the value inside the function, please consider that as correct.

[b] A function to swap two integers using 'call-by-reference'. (2.5 marks)

```

// Function to swap two integers
void swap(int *a, int *b) {
    int temp = *a; // Store the value at address a in temp
    *a = *b;        // Assign value at address b to address a
    *b = temp;      // Assign value of temp to address b
}

```

If the return type is different and they have printed the value inside the function, please consider that as correct.