
SURPRISE TEST 1 SOLUTIONS (SECTION A) MAX: 2 MARKS. TIME:
20 MINUTES

NAME: _____ ROLL NUMBER: _____

Note: Negative mark of -1 applies for each incorrect or incomplete response.

Question 1. I was writing a program to find the length of a string without using the strlen function. But, a mischievous friend of mine deleted one line. Can you fill in the blanks to complete this code and make it work? Assume that the input string will have shorter than 100 characters. [1 Mark]

```
int mystrlen(char *str)
{
    static int length=0;
    if(*str!=NULL)
    {
        length++;
        mystrlen(++str) ; //The answer is here.
    }
    else
    {
        return length;
    }
}
int main()
{
    char str[100];
    int length=0;

    printf("Enter a string: ");
    gets(str);

    length=mystrlen(str);

    printf("Length = %d\n",length);

    return 0;
}
```

Explanation: This code is particularly interesting because all these days we were using `strlen` to conveniently find the length of a string. Here, we use a simple idea that we can iterate through the characters of the string and each time when we iterate, we keep a count. This count gives us the length of the string. A loop could have also been used. But, we could do the same using one line of code. We recurse here!

Question 2. What is the output? [1 Mark]

```
void fun(int arr[], int d, int n)
{
    int i;
    for (i = 0; i < d; i++)
        funByOne(arr, n);
}

void funByOne(int arr[], int n)
{
    int temp = arr[0], i;
    for (i = 0; i < n - 1; i++)
        arr[i] = arr[i + 1];
    arr[i] = temp;
}

void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
}

int main()
{
    int arr[] = { 1, 2, 3, 4, 5, 6, 7 };
    fun(arr, 4, 7);
    printArray(arr, 7);
    return 0;
}
```

Answer = 5 6 7 1 2 3 4

Explanation: Arrays are always passed by reference. `funByOne` method takes an array as input and rotates its elements. The logic to rotate is interesting. First we store the first element in a temp variable. Then shift all array elements by one position. Finally, we put the temp element in the last position. See `funByOne` method in the code above. `printArray` method is simply printing the array. In this code, we are rotating the array `d` times where `d` is 4. Therefore, the output is 5 6 7 1 2 3 4.