

CS10003 Programming and Data Structures

Spring Semester, 2021-2022

Sections 1 & 2

Long Test 1 – Part A [Marks = 50]

18-May-2022, 9:00 to 10:10

---

**INSTRUCTIONS**

- You have 1 hour for writing and 10 minutes for submission.
  - Write your answers on paper. Answers must be handwritten. Typed or written answers using an electronic device are not allowed.
  - Write your name and roll number on each page. Write page numbers for each page.
  - Scan all pages and collate. Create a single PDF file (of size  $\leq 10$  MB). You could also take pictures of different pages, combine them to make a single pdf file.
  - The name of the files for parts A should be <RollNumber>\_Long1A. Ensure your roll number in the filename has only digits and uppercase characters.
  - Upload your file. Make sure you click on the 'Turn in' button to submit your file.
  - The said deadlines are strict after which no submissions will be allowed.
  - We will not accept submission by any other means.
- 

1. Consider the following program.

```
int main() {
    int a = 13, b = 5, c = 7;
    float x; int y; char z;

    x = 4*4/3*a/b;
    printf("X=%3.2f",x);           (A)

    y = 3.0*b/c/a;
    printf("Y = %d",y);           (B)

    a += a*a%2; a %= 2;
    (a)? printf("Good") : printf("Nice");   (C)

    return (0);
}
```

What value the program will print at (A), (B) and (C)?

10

**Solution:**

- (A) X= 13.00
- (B) Y= 0
- (C) Nice

2. We are given eight integers in the input to a C program, out of which all except a single one is distinct. We wish to print the value of the unique distinct integer and also determine the variable in the input order corresponding to the distinct value. Use a single scanf statement to read all the eight integers, say,  $a, b, c, d, e, f, g$  and  $h$ . Do not use arrays. For example, if the sequence is 1 1 1 1 1 2 1 1 then we must print that 2 is the unique distinct value.

10

**Solution:**

```
#include<stdio.h>

int main() {
    int a,b,c,d,e,f,g,h;
    scanf("%d %d %d %d %d %d %d %d", &a, &b, &c, &d, &e, &f, &g, &h );

    if (a+b==c+d) {
        if (e==f) {
            if (e==g) printf ("8th number (%d) is distinct\n",h);
            else printf("7th number (%d) is distinct\n",g);
        }
        else {
            if (e==g) printf ("6th number (%d) is distinct\n",f);
            else printf("5th number (%d) is distinct\n",e);
        }
    }

    else {
        if (a==b) {
            if (c==a) printf ("4th number (%d) is distinct\n",d);
            else printf("3rd number (%d) is distinct\n",c);}
        else {
            if (a==c) printf ("2nd number (%d) is distinct\n",b);
            else printf("1st number (%d) is distinct\n",a);
        }
    }
    return 0;
}
```

3. Write a C program to read in a positive integer  $n$  and find the digit which occurs the maximum number of times in  $n$  along with its frequency. For example, if the input is 567755, then the answer is 5 and its frequency is 3. Do not use arrays or any library functions other than standard input and output. There should be only one main function and no other functions defined by you.

10

**Solution:**

```
#include<stdio.h>

main()
{
    int m, n, maxdigit, maxfreq, tempfreq, i, j;
    scanf("%d", &n);
    printf("n = %d \n", n);
    maxdigit = -1;
    maxfreq = 0;
    for (i = 0; i < 10; i++)
    {
        m = n;
        tempfreq = 0;
        while (m > 0)
```

```

    {
        if (m%10 == i) tempfreq++;
        m = m/10;
    }
    if (tempfreq > maxfreq)
    {
        maxfreq = tempfreq;
        maxdigit = i;
    }
}
printf("Digit %d has maximum frequency of %d \n", maxdigit, maxfreq);
}

```

4. Write a program which reads a list of 10 numbers in ascending order. It then prints the pairs of consecutive numbers having maximum and minimum differences respectively. Your program should have only one *main()* function and no other functions. 10

**Solution:**

```

#include <stdio.h>

int main(){
    int i, diff, maxdiff = 0, mindiff, k1, k2, mink1, mink2, maxk1, maxk2;
    scanf("%d", &k1);
    scanf("%d", &k2);
    mindiff = diff = k2 - k1;
    maxk1 = mink1 = k1;
    maxk2 = mink2 = k2;
    for(i=2; i<10; i++){
        k1 = k2;
        scanf("%d", &k2);
        diff = k2 - k1;
        if(diff < mindiff){
            mindiff = diff;
            mink1 = k1;
            mink2 = k2;
        }
        if(diff > maxdiff){
            maxdiff = diff;
            maxk1 = k1;
            maxk2 = k2;
        }
    }
    printf("Pair with minimum difference: %d %d\n", mink1, mink2);
    printf("Pair with maximum difference: %d %d\n", maxk1, maxk2);
    return 0;
}

```

5. A group of students receive grades for a subject and you are interested in seeing the grade distribution. Write a program that takes as input the number of students  $n$  and then reads  $n$  grades (integers in the set  $\{1, 2, 3, 4, 5\}$ ) into an array  $A[0, \dots, n-1]$ . The output should be a bar graph with bars represented by sequence of '#' characters. You are allowed to use constant amount of additional storage. Assume  $1 \leq n \leq 60$ . 10

### Sample Input/Output

```
Enter n (<=60) : 10
Enter 10 grades: 2 3 4 5 2 2 1 3 3 3
#
##
##
#####
```

### Solution:

```
#include <stdio.h>

int main(){
    int n, i, j, max = -1;
    int A[60], B[6] = {0,0,0,0,0,0};

    printf("Enter n (<=60): ");
    scanf("%d", &n);
    if (n >60 || n <1){
        printf("n should be positive and at most 60.\n\n");
        return 1;
    }

    printf("Enter %d grades: ", n);

    for(i=0; i<n; i++){
        scanf("%d", &A[i]);
        if (A[i] < 1 || A[i] > 5){
            printf("Array elements must be positive and at most 5.\n\n");
            return 2;
        }
        B[A[i]]++;
    }

    max = B[1];
    for(i=2; i<=5; i++)
        if(B[i] > max)
            max = B[i];

    for(j=max; j>=1; j--){
        for(i=1; i<=5; i++){
            if(B[i] >= j)
                printf("#");
            else
                printf(" ");
        }
        printf("\n");
    }

    return 0;
}
```