

CS10003 Programming and Data Structures

Spring Semester, 2021-2022

Sections 3 & 4

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 part 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. Write a function *readinteger()* that reads a sequence of characters skipping all non-numeric digits. It then stores the number formed by the numeric characters in a variable of type long int and returns it. For example, if 234rttrre679y357u864dah is the input stream then it stores the integer 234679357864 as a long int number, and returns it.

Write the *main()* program to do the following in exactly the sequence as stated below.

- Use the *readinteger()* function to read two integers.
- Compute the sum and product of these two numbers and print them.

10

**Solution:**

```
#include<stdio.h>

long int readinteger() {
    char ch; long int x=0; // ch for input of characters, x is returned

    scanf("%c",&ch);
    while (ch!='\n') {
        if (ch>=48&&ch<=57) {
            x=10*x+ch-48;
            scanf("%c",&ch);
        }
    }
    return x;
}
```

```

}

int main(){
    long int num1=readinteger();
    long int num2=readinteger();
    long int sum=num1+num2;
    long int prod=num1*num2;
    printf("Sum of the integers read = %ld\n",sum);
    printf("Product of the integers read = %ld\n",prod);
    return 0;
}

```

2. Read three floating point coordinates of points  $A = (x_1, y_1)$ ,  $B = (x_2, y_2)$ ,  $C = (x_3, y_3)$  of points in a 2-D plane. Assume the points are not collinear. Determine if the Triangle  $ABC$  is equilateral, isosceles or neither. Do not use arrays or loops. You may include and use the `math.h` library in addition to standard input and output. There should be only one main function and no other functions defined by you. 10

**Solution:**

```

#include<stdio.h>
#include<math.h>
main()
{
    float x1, y1, x2, y2, x3, y3;
    float AB, AC, BC;
    scanf("%f%f%f%f%f", &x1, &y1, &x2, &y2, &x3, &y3);
    printf("A = <f, %f>, B = <f, %f>, C = <f, %f> \n", x1, y1, x2, y2, x3, y3);
    AB = sqrt(pow(x1-x2, 2) + pow(y1-y2,2));
    AC = sqrt(pow(x1-x3, 2) + pow(y1-y3,2));
    BC = sqrt(pow(x2-x3, 2) + pow(y2-y3,2));
    printf("AB = %f, AC = %f, BC = %f \n", AB, AC, BC);
    if ((AB == AC) && (AB == BC))
        printf("Equilateral Triangle");
    else if ((AB == AC) || (AB == BC) || (AC == BC))
        printf("Isosceles Triangle");
    else printf("Neither Equilateral nor Isosceles");
}

```

3. S. Ramanujan was an Indian mathematician who became famous for his intuition for numbers. When the English mathematician G. H. Hardy came to visit him in the hospital one day, Hardy remarked that the number of his taxi was 1729, a rather dull number. To which Ramanujan replied, "No, Hardy! No, Hardy! It is a very interesting number. It is the smallest number expressible as the sum of two cubes in two different ways." In order to verify this claim, write a function that takes an integer command-line argument  $n$  and prints all integers less than or equal to  $n$  that can be expressed as the sum of two cubes in two different ways - find distinct positive integers  $a$ ,  $b$ ,  $c$ , and  $d$  such that  $a^3 + b^3 = c^3 + d^3$ . Use four nested for loops. 10

**Solution:**

```

#include <stdio.h>

int main(){
    int n;
    scanf("%d", &n);
    int a,b,c,d;

    for(a=1; a<=n; ++a)

```

```

    for(b=a+1; b<=n; ++b)
        for(c=a+1; c<=n; ++c)
            for(d=c+1; d<=n; ++d)
                if(a*a*a+b*b*b == c*c*c+d*d*d)
                    printf("(%d %d %d %d)\t Sum of cubes = %d\n", a,b,c,d,a*a*a+b*b*b);

    return 0;
}

```

4. Call a number curious if it is divisible by its *most significant digit* (leftmost non-zero digit). For example, 489 is not divisible by 4 and hence is not curious. On the other hand, 488 is curious as it is divisible by 4. Write a C program that reads two integers  $a, b$  with  $0 < a < b$  and prints all curious numbers in the range  $[a, b]$ . Your program should have only one `main()` function and no other functions. Do not use any library functions other than standard input/output. 10

**Solution:**

```

#include <stdio.h>

int main(){
    int a, b;
    int msb, i, k, flag;
    printf("Enter 2 integers: ");
    scanf("%d %d", &a, &b);
    flag = 0;
    for(i=a; i<=b; i++){
        k = i;
        while(k > 0){
            msb = k%10;
            k = k/10;
        }
        if(i%msb == 0){
            flag = 1;
            printf("%d ", i);
        }
    }
    if(flag == 0)
        printf("No curious numbers in the given range.\n\n");
    else
        printf("\n\n");
    return 0;
}

```

5. Fill in the blanks to complete the following program that reads a string  $S$  as input (assume that the string can be at most 50 characters long and will contain only alphabetic letters (lower case and upper case)), and forms a new string  $R$  that contains, in lower case and in order, all letters (from the alphabet) that are not present in  $S$ , in either lower case or upper case, and finally, prints  $R$ . For example, if  $S = \text{"rtaBfFHgDceAPQ"}$ , then  $R = \text{"ijklmnosuvwxzyz"}$ .

```

int main() {
    char S[100], R[100], temp;
    int i, j, X[26];

    scanf("%s", S);    // Read the input from user

    for(i = 0; i < 26; i++) X[i] = 0;
}

```

```

    for ( _____ ) { (A)
        if (S[i] >= 'A' && S[i] <= 'Z')
            temp = _____ ; (B)
        else
            temp = S[i];
        X[ _____ ] = 1; (C)
    }

    /* Write to R */
    j = 0;
    for ( i = 0; i<26; i++ )
        if (X[i] == 0) { _____ } (D)
            R[j] = '\0';
        printf("R = %s\n", R);
    return 0;
}

```

**Solution:**

(A) `i = 0; S[i] != '\0'; i++`

(B) `'a' + S[i] - 'A'`

(C) `temp - 'a'`

(D) `R[j++] = 'a' + i;`