

CS10003 Programming and Data Structures

Spring Semester, 2021-2022

Sections 7 & 8

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 ≤ 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 program that reads three points, (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) and determines whether they represent the vertices of a right-angled triangle. If so, the program must identify the hypotenuse, and print the value of one of the angles in radians. You may use appropriate library functions, but please include the necessary header files. Your program should have only the `main()` function.

10

Solution:

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

int main(){
    double x1,x2,x3,y1,y2,y3,s12,s23,s13;

    printf("Enter coordinates of 3 points: ");
    scanf("%lf %lf %lf %lf %lf %lf",&x1,&y1,&x2,&y2,&x3,&y3);

    s12 = (x2-x1)*(x2-x1)+(y2-y1)*(y2-y1);
    s23 = (x3-x2)*(x3-x2)+(y3-y2)*(y3-y2);
    s13 = (x1-x3)*(x1-x3)+(y1-y3)*(y1-y3);

    if(s12 + s13 == s23){
        printf("Triangle formed by the 3 points is right angled.\n");
        printf("Hypotenuse is the line segment joining (%lf,%lf) and (%lf,%lf).\n\n", x2,y2,x3,y3);
        printf("One of the other 2 angles (in radians) = %lf\n\n",asin(sqrt(s12/s23)));
    }
    else if(s12 + s23 == s13){
```

```

    printf("Triangle formed by the 3 points is right angled.\n");
    printf("Hypotenuse is the line segment joining (%lf,%lf) and (%lf,%lf).\n\n", x1,y1,x3,y3);
    printf("One of the other 2 angles (in radians) = %lf\n\n",asin(sqrt(s12/s13)));
}
else if(s23 + s13 == s12){
    printf("Triangle formed by the 3 points is right angled.\n");
    printf("Hypotenuse is the line segment joining (%lf,%lf) and (%lf,%lf).\n\n", x1,y1,x2,y2);
    printf("One of the other 2 angles (in radians) = %lf\n\n",asin(sqrt(s23/s12)));
}
else
    printf("Triangle formed by the 3 points is not right angled.\n\n");

return 0;
}

```

2. A giant is a person who is simultaneously the tallest and heaviest among a set of people. Write a program that, given the heights and weights of 3 people $(h_1, h_2, h_3, w_1, w_2, w_3)$, outputs whether any of them is a giant. The i -th person is a giant if $h_i \geq h_j$ and $w_i \geq w_j$ for all $j \in \{1, 2, 3\}, j \neq i$. Assume all heights and weights are positive floating point numbers. Your program should use only the ternary conditional operator $?:$ and contain no if-else statements. Your program should have only the `main()` function. Do not use any library functions other than standard input and output.

Sample Output 1

```

Heights of persons 1,2,3: 4.5 5.8 5.7
Weights of persons 1,2,3: 50 90 100
There is no giant.

```

Sample Output 2

```

Heights: 4.7 6 5.8
Weights: 50 102 81
Person 2 is a giant.

```

10

Solution:

```

#include <stdio.h>

int main(){
    float h_1,w_1,h_2,w_2,h_3,w_3;
    printf("Heights of persons 1,2,3: ");
    scanf("%f %f %f", &h_1,&h_2,&h_3);
    printf("Weights of persons 1,2,3: ");
    scanf("%f %f %f", &w_1,&w_2,&w_3);

    int giantIndex;

    giantIndex = (h_1 >= h_2 && h_1 >= h_3 && w_1 >= w_2 && w_1 >= w_3)? 1 :
    ((h_2 >= h_3 && h_2 >= h_1 && w_2 >= w_3 && w_2 >= w_1) ? 2 :
    ((h_3 >= h_2 && h_3 >= h_1 && w_3 >= w_2 && w_3 >= w_1)?3:0));

    (giantIndex)?printf("Person %d is a giant.\n",giantIndex):printf("There is no giant.\n");
    return 0;
}

```

3. A number is an Armstrong number, if the sum of the cubes of its digits is equal to the number itself. For example, $157 = 1^3 + 5^3 + 7^3$ and $371 = 3^3 + 7^3 + 1^3$, etc. are Armstrong numbers. The following program is intended to determine whether an input number n is an Armstrong number. Fill the blanks to complete the statements so that it does the job. Explain the statements you add.

```
int main() {
    int n, temp, x, sum;
    scanf("%d",&n);
    temp=n;
    while(n>0) {
        x = _____;          (A)
        sum += _____;      (B)
        n = _____;         (C)
    }
    (temp==sum) ? printf("Yes"): printf("No");

    return 0;
}
```

10

Solution:

(A) $n\%10$
 (B) $x*x*x$
 (C) $n/10$

4. Write a C program to compute and print the mean and variance of $3n$ integers read 3 at a time. The program must read n and then the n triples sequentially using only a constant number of variables, no arrays. After reading a triple, the program must print the mean and variance for that triple. After all $3n$ integers are read, the program should print the mean and variance of all the integers. There should be only one *main()* function and no other function.

10

Solution:

```
#include <stdio.h>

int main (){

    int a, b, c, n,i;
    float mean, variance, gmean, gvariance;

    scanf("%d",&n);

    gvariance=0;
    gmean=0;
    for (i=0;i<n;i++) {
        scanf("%d%d%d", &a, &b, &c);
        mean =(a+b+c)/3.;
        variance=((a-mean)*(a-mean)+(b-mean)*(b-mean)+(c-mean)*(c-mean))/3.;
        printf("Triple %d: Mean = %f Variance= %f \n", i,mean,variance);

        gvariance+= a*a+b*b+c*c;
        gmean=(gmean*(i)*3+mean*3)/(3*(i+1));
    }
}
```

```

    gvariance = gvariance/(3*n) - gmean*gmean;
    printf("\nFor all numbers read: Mean = %f Variance= %f \n", gmean,gvariance);
}

```

5. Write a C program to read in a character string S (which does not have any blanks in between) and remove all vowels and print the string P obtained after removal of all vowels in S . Assume that the strings S and P will have at most 30 characters. You should handle both lowercase and uppercase vowels for removal. For example, if the input string is “compUTers”, the output should be “cmpTrs”. Do not use any library functions other than standard input and output. You can use arrays. There should be only one main function and no other functions defined by you.

10

Solution:

```

#include<stdio.h>
main()
{
    char S[31], P[31];
    char vowels[] = {'a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U'};
    int i, j, k, len, found;
    k = 0;
    len = 0;
    scanf("%s", S);
    while (S[len]!='\0') len++;
    printf("S = %s, Length of S = %d \n",S, len);
    for (i = 0; i < len; i++) {
        found = 0;
        for(j=0; j< 10; j++)
        {
            if(S[i]== vowels[j])
            {
                found = 1;
                break;
            }
        }
        if (found == 0) P[k++] = S[i];
    }
    P[k] = '\0';
    printf("P = %s, Length of P = %d \n", P, k);
}

```