National University of Computer & Emerging Sciences, Karachi
Fall -2021 (School of Computing)
Final Exam, 09:00 am - 12:00 pm, Wed 29th December, 2021

Course Code: CS1002	**Course Name: Programming Fundamentals**
Instructors: Dr. Abdul Aziz, Dr. Farrukh Shahid, Dr. Murk Marvi, Mr. Basit Ali, Ms.Abeer Gauher, Mr. Hamza Ahmed, Ms. Attiya Jokio, Ms. Sobia Iftikhar. Ms. Sumaiya.	
Student ID:	**Section:**

Instructions:

- Attempt all questions on answer sheet, including MCQ's.
- Return the question paper and make sure to keep it inside your answer sheet.
- Read each question completely before answering it. There are six questions and six pages.
- In case of any ambiguity, you may make assumption. However, your assumption should not contradict any statement in the question paper
- Do not write anything on the question paper (except your ID and group).

Total Points: 100

Question 1: Multiple choice questions. **(1 mark each= 10 marks)**

1. Maximum number of elements in the array declaration *'int a[5][8]'* are:
   a) 28 b) 32 c) 35 d) 40

2. A pointer to a pointer is a form of:
   a) Multiple indirections
   b) A chain of pointers
   c) Both a and b
   d) None of these

3. An expression contains relational, assignment and arithmetic operators. If Parenthesis are not present, the order will be:
   a) Assignment, arithmetic, relational
   b) Relational, arithmetic, assignment
   c) Assignment, relational, arithmetic
   d) Arithmetic, relational, assignment

4. p++ executes faster than p+1 because:
   a) p uses temparory memory
   b) p++ is a single instruction
   c) ++ is faster than +
   d) None of these

5. ***"int testarray[3][2][2] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};"*** What value does **testarray[2][1][0]** in the sample code above contain?
   a) 11 b) 7 c) 5 d) 9

6. What is printed when the sample code above is executed?

```
int y[4] = {6, 7, 8, 9};
int *ptr = y + 2;
printf("%d ", ptr[ 1 ] );
```

   a) 6 b) 7 c) 8 d) 9

7. What is printed when the sample code above is executed?

```
#include <stdio.h>
void reverse(int i);
int main()
{    reverse(1);  }
 void reverse(int i)
  {     if (i > 5)
              {return ;}
        reverse((i++, i));
        printf("%d ", i);   }
```

   a) 1 2 3 4 5
   b) 5 4 3 2 1
   c) Compilation error
   d) 6 5 4 3 2

8. What is printed when the sample code above is executed?

```
#include <stdio.h>
int main()
{
   int x = 10, *y, **z;

   y = &x;
   z = &y;
   printf("%d %d %d", *y, **z, *(*z));
   }
```

a) 10 10 10

b) 100xaa54f10

c) Run time error

d) No Output

9. Which of the following is executed by Preprocess?

a) #include<stdio.h>

b) return 0

c) void main(int argc , char ** argv)

d) None of above

10. If x is an array of integer, then the value of &x[i] is same as:

a) &x[i-1] + sizeof (int)

b) x + sizeof (int) * i

c) x+i

d) none of these

Question 2. Show the output. **(1 mark each= 10 marks)**

What will be the output?

1.
```
main(){
        char *p="Hello world";
        int *q;
        p++;
        q = (int *)p;
        q++;
        printf("%s%s",p,q); }
```

2.
```
main()
{
int color=2;
switch(color)
{
        case 0: printf("Black");
        case 1: printf("Blue");
        case 2: printf("Green");
        case 3: printf("Aqua");
        default: printf("Other");
} }
```

3.
```
main()

{
char s[ ]="man";
int i;
for(i=0;s[ i ];i++)
printf("%c%c%c%c",s[ i ],*(s+i),*(i+s),i[s]);
}
```

4.
```
main()

{
        int  const * p=5;
        printf("%d",++(*p));
}
```

5. ***main()***
```
 {
  int array[10] = {3, 0, 8, 1, 12, 8, 9, 2, 13, 10};
  int x, y, z;
 x = ++array[2];
 y = array[2]++;
 z = array[x++];
 printf("%d %d %d", x, y, z);
 }
```

6. ***int main()***
```
{
int arri[] = {1, 2 ,3};
int *ptri = arri;
char arrc[] = {1, 2 ,3};
char *ptrc = arrc;
printf("sizeof arri[] = %d ", sizeof(arri));
printf("sizeof ptri = %d ", sizeof(ptri));
printf("sizeof arrc[] = %d ", sizeof(arrc));
printf("sizeof ptrc = %d ", sizeof(ptrc)); }
```

7. ***main()***
```
 {
 int val = 1;

 do {
     val++;
     ++val;
 } while (val++ > 25);
 printf("%d\n", val);}
```

8. ***main()***
```
 {
 char str1[] = "Hello. How are you?";
 char str2[21];
 int pos;
 for(pos=0; pos<21; pos++);
{
  str2[pos] = str1[pos];
}
printf("str1: %s, str2: %s", str1, str2); }
```

9. ***main()***
```
 {
 char *_ptr = malloc(100);
 if(_ptr!=NULL)
 free(_ptr);
 free(_ptr); }
```

10. ***main()***
```
 {
      int x = 5,  y = 10;
      int const *p = &x;
      *p = 15;
      printf("%d", x); }
```

Question 3. Ali is playing a game of cards. This is a special type of cards in which cards are a 3 digit number (i.e. 100 - 999). He has to sort the cards in ascending order. However, sorting will be done on the sum of the digits. Make a c program to sort his cards, take user input for number of cards. Taking input of cards number should be a function named **get_data()**. Sorting should be another function **sort()** and displaying the final result should be another function **display()**. **(20 marks)**

Expected Input:

```
How many cards do you have? 5
Enter number of card 1:    291
Enter number of card 2:    124
Enter number of card 3:    371
Enter number of card 4:    574
Enter number of card 5:    189
```

Expected Output:

```
Sorted card 1 : 124
Sorted card 1 : 371
Sorted card 1 : 291
Sorted card 1 : 574
Sorted card 1 :189
```

Question 4. Someone has asked you to design a special purpose calculator with the following two options.

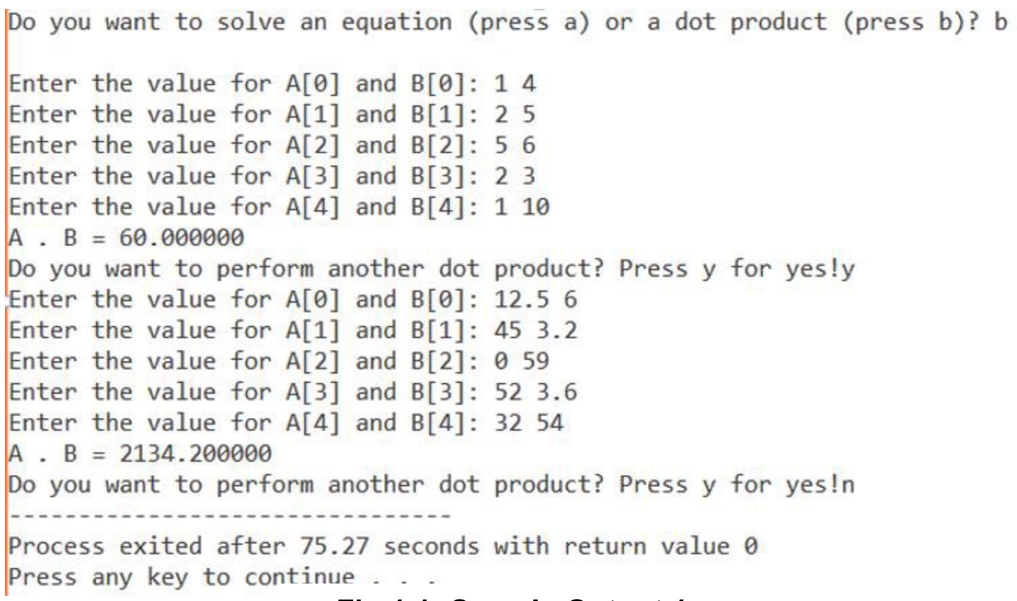
a. Solve an equation involving combination of four (+, -, *, /) operators and maximum 100 operands. Assume that all the operators have same precedence and equation is supposed to be solved from left to right. For example, 3+34.5/2*60-22 is an equation, and it consists of six operands and five operators. In order to store all the operands and operators of the equation, define an array of type "**struct data**" having three members; two for operands and one for operator. At the first index of the array store the first two operands (3), (34.5) and the first operator (+). At the next index of the array, store the result of the last operation (37.5), the next operator (/), and the next operand (2). Call a user-defined function **basic( )** in order to perform a given arithmetic operation on two operands. Continue this until all the operands and operators of the equation are entered and you get the final result.
b. Obtain the dot product of two vectors of same size. Note that if A=[1,2,5,2,1] and B=[4,5,6,3,10], then the dot product can be obtained as A . B = 1*4 + 2*5 + 5*6 + 2*3 + 1*10 = 60.

In main (), ask the user whether she wants to solve an equation or dot product. The user should press **a** for choosing equation and **b** for choosing dot product. If the user enters any character other than **a or b** the program should terminate after printing an invalid input message.

If the user chooses **a**, then ask how many operands are there in the equation. Store the operands and operators entered by the user in an array of type **struct data.** In case the user enters an invalid operator or operands, print the message of invalid input. Otherwise, call a user defined function **basic( )** to get the required result. Print the result by calling **display( )** function from **main( )**. The program ends only if the user wants to terminate it.
If the user chooses dot product by pressing **b**, then ask her to enter the elements of two arrays. Call **dot( )** function in order to calculate the dot product of two arrays. Print the result in **main()**. The program ends only if the user wants to terminate it. **(20 marks)**

For defining **basic( )**, **display( )**, and **dot( )** functions, please follow the following details.

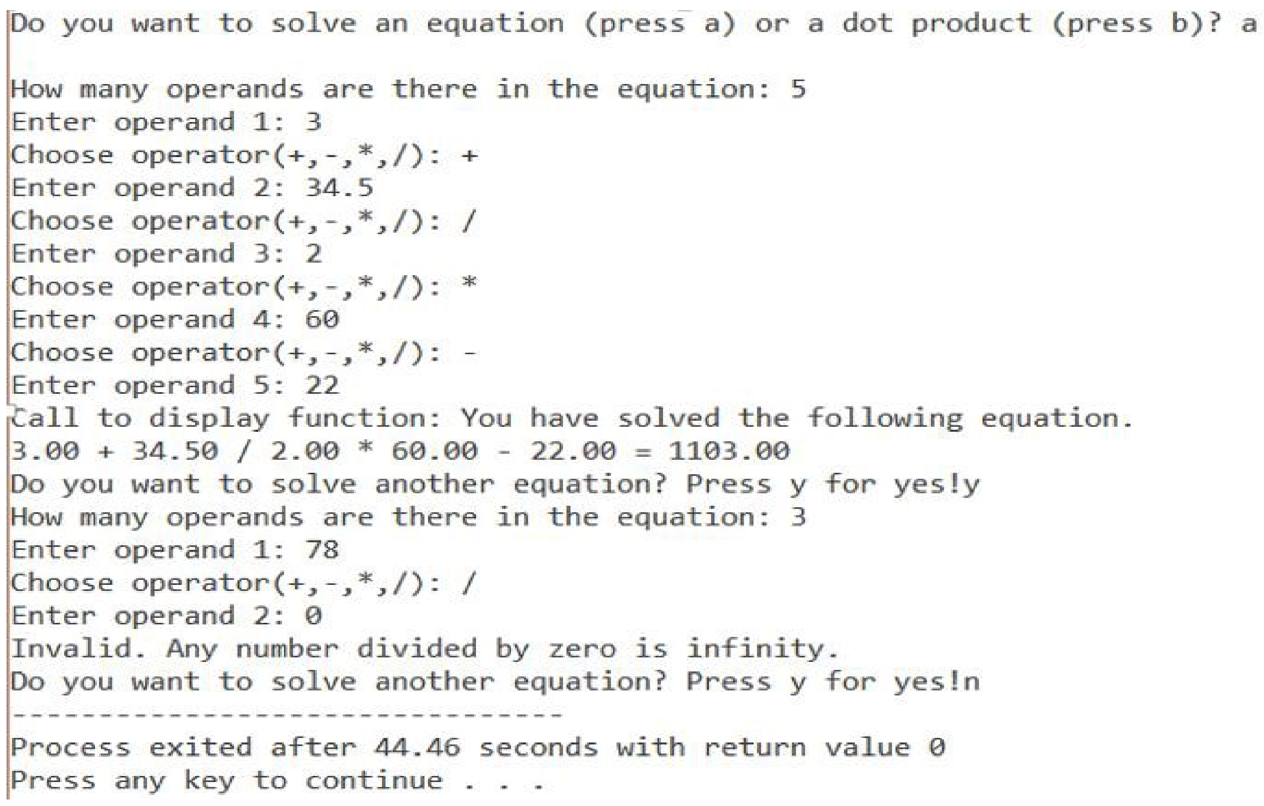
a. **float basic(float, float, char op);**
b. **void display(const struct data [ ], int);**
c. **double dot(const double [ ], const double [ ], int);**



```
Do you want to solve an equation (press a) or a dot product (press b)? b

Enter the value for A[0] and B[0]: 1 4
Enter the value for A[1] and B[1]: 2 5
Enter the value for A[2] and B[2]: 5 6
Enter the value for A[3] and B[3]: 2 3
Enter the value for A[4] and B[4]: 1 10
A . B = 60.000000
Do you want to perform another dot product? Press y for yes!y
Enter the value for A[0] and B[0]: 12.5 6
Enter the value for A[1] and B[1]: 45 3.2
Enter the value for A[2] and B[2]: 0 59
Enter the value for A[3] and B[3]: 52 3.6
Enter the value for A[4] and B[4]: 32 54
A . B = 2134.200000
Do you want to perform another dot product? Press y for yes!n
--------------------------------
Process exited after 75.27 seconds with return value 0
Press any key to continue . . .
```

Fig 1.1: Sample Output 1



```
Do you want to solve an equation (press a) or a dot product (press b)? a

How many operands are there in the equation: 5
Enter operand 1: 3
Choose operator(+,-,*,/): +
Enter operand 2: 34.5
Choose operator(+,-,*,/): /
Enter operand 3: 2
Choose operator(+,-,*,/): *
Enter operand 4: 60
Choose operator(+,-,*,/): -
Enter operand 5: 22
Call to display function: You have solved the following equation.
3.00 + 34.50 / 2.00 * 60.00 - 22.00 = 1103.00
Do you want to solve another equation? Press y for yes!y
How many operands are there in the equation: 3
Enter operand 1: 78
Choose operator(+,-,*,/): /
Enter operand 2: 0
Invalid. Any number divided by zero is infinity.
Do you want to solve another equation? Press y for yes!n
--------------------------------
Process exited after 44.46 seconds with return value 0
Press any key to continue . . .
```

Fig 1.2: Sample Output 2

Question 5. Suppose we have the following character array: **Char test [ ] ="This is a test string for my program"**

Required: You are required to write a code for the following operations: **(5 marks each = 20 marks)**

a. Reverse the given string without using any additional array or even an extra variable. The resultant array will look like "margorp ym rof gnirts tset a si sihT".

b. Write a function merge (char [ ], start position, "text to be insert")

c. Wildcard Pattern Matching: Given a string and a pattern containing wildcard characters, i.e., `*` and `?`, where `?` Can match to any single character in the string and `*` can match to any number of characters including zero characters, design an efficient algorithm to check if the pattern matches with the complete string or not.

Input: string = "xyxzzxy", pattern = "x***y"

Output: Match

Input: string = "xyxzzxy", pattern = "x***x"

Output: No Match

Input: string = "xyxzzxy", pattern = "x***x?"

Output: Match

d. Write a recursive function to reverse the input words.

Input: "This is a test string for my program"

Output: "program my for string test a is This"

Question 6. Note the following: [DoB = Date of Birth, LSDHN= Least Significant Digit of your House no / Flat no, LSDSID = Least Significant Digit of your Student ID, MSDBY = Most Significant Digit of your Birth Year, BM= Birth Month]. Consider a Vehicle to Vehicle (V2V) scenario as shown in the figure a, containing, three autonomous cars (AC) AC1, AC2, AC3, a jammer J, a cloud server containing control unit (CU), four locations A, B, C, and D. It is assumed that AC cars pass the four locations on any given day of a week in sequential order (AC1 passes first, followed by AC2 and AC3). All cars are connected with the cloud based remote serve and exchange information such as speed, location, direction, weather condition, steering angle, etc. with the CU in the server using wireless link (Wi-Fi or 5G Cellular network). It is known that wireless link is susceptible to the jamming attacks and jammer tries to alter the information of the AC by accessing the values. For simplicity, assume that AC cars send speed information measured at locations A, B, C and D to the CU in the server. The CU stores the velocity information collected from all cars in the following given format.

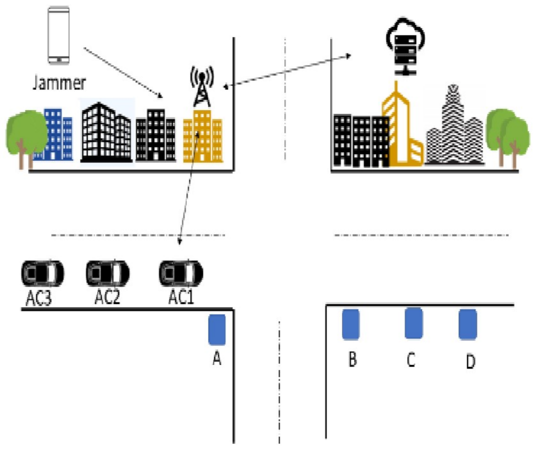


Fig.2. V2V Scenario

$$AC_{speed\_information} = \begin{array}{c} \\ \text{AC1} \\ \text{AC2} \\ \text{AC3} \end{array} \begin{array}{cccc} \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} \\ \end{array}$$

$$AC_{speed\_information} = \begin{array}{cc} & \begin{array}{cccc} \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} \end{array} \\ \begin{array}{c} \text{AC1} \\ \text{AC2} \\ \text{AC3} \end{array} & \begin{bmatrix} BM & 6 & 10 & LSDHN \\ 3 & 4 & LSDSID & 23 \\ TD & 45 & 100 & MSDBY \end{bmatrix} \end{array}$$

The jammer attacks the AC1, AC2 or AC3 on the speed values as, $JAC1 = [1 \quad 0 \quad 12 \quad 100]$, $JAC2 = [100 \quad 110 \quad 12 \quad 100]$, $JAC3[11 \quad 10 \quad 112 \quad 100]$ and the overall effect is,

$$\boldsymbol{Jammerattack} = [\boldsymbol{JAC1}, \boldsymbol{JAC2}, \boldsymbol{JAC3}]^{\boldsymbol{T}}$$

The jammer attack has additive effect on the speed values given as,

$$\boldsymbol{Resultant} = \boldsymbol{AC}_{\boldsymbol{speed\_information}} + \boldsymbol{Jammerattack}$$

Develop a C-code which store the above given information of the speed and jammer attack and calculate the Resultant. The program should determine the average of the resultant and if resultant is greater than your DOB, it should display the message Jammer attack is devastating otherwise display Jammer attack is less harmful. **(20 marks)**