TUTORIAL 5 (SECJ1013)

PROGRAMMING TECHNIQUE 1

SECTION 04 & 07, SEM 1, 2024/2025

Q#1

Write C++ code for each task below. Note that questions 4(a) – (e) are continuous.

a) Declare and initialize a 6-element array named arr with the following given list of values:

156, -25, 24

b) Calculate the difference between the first and fifth elements of the array and assign the value to the last element.

c) Using an appropriate loop, print the contents of all elements in arr.

d) Using an appropriate loop, search for and print the minimum value in arr.

e) Using appropriate loops, reverse the array arr so that the content of the first element goes to the last element and vice versa, the content of the second element goes to the second last element and vice versa, and so forth.

Q#2

A C++ program that uses and manipulates a two-dimensional integer array of fifty rows and twenty columns is going to be written. Write an appropriate C++ code segment that performs each of the following tasks:

a) Declare the array.

b) Fill in the array with numbers entered by the user.

c) Display the numbers of the first and last columns for each row.

d) Find the row index of the element in the first column that contains the largest number.

e) Based on the row index found in (d), display the largest number.

Q#3

Given a two-dimensional array as declared in the Program 1 below. Based on the program, answer the questions following it.

```
1  //Program 1
2  #include <iostream>
3  #include <cmath>
4  using namespace std;
5
6  #define NROW 2
7  #define NCOL 3
```

```
8
9   int main(){
10     int numbers[NROW][NCOL];
11
12     for (int i=0; i<NROW; ++i)
13        for (int j=0; j<NCOL; ++j)
14           numbers[i][j] = pow((i+j),2.0);
15
16     for (int i=0; i<NROW; ++i){
17        for (int j=0; j<NCOL; ++j)
18           cout << numbers[i][j]*2 << "\t";
19        cout << endl;
20     }
21     return 0;
22  }
```

a) Write the output printed by the program.

b) Define a function named calculate to calculate and print the square root of the sum of all the elements in an array. The function should accept a one-dimensional array as its parameters.

c) Then, in the main function, write a statement that will invoke the function calculate to calculate and print the square root of the sum of all the elements for each row in the array numbers.

Q#4

Program 2 below is incomplete and it does not include the definitions of functions **`inputData`**, **`compare`** and **`compute`**. Complete the definition of each of these functions according to the requirements stated as comments in the next page. Calls to these functions are shown in the main function.

```
1   //Program 2
2   #include <iostream>
3   using namespace std;
4   #define SIZE 5
5
6   void inputData(int[]);
7   void compare(int [],int [] );
8   int compute(int [],int []);
9
10  int main() {
11     int arr1[SIZE],sum=0;
12     int arr2[SIZE]={2, 3, 2, 2, 1};
13
14     inputData(arr1); compare(arr1,arr2);
15     sum = compute(arr1,arr2);
16     cout << "Sum of elements in new array :" << sum << endl;
```

```
17    return 0;
18  }
```

a) Function **`inputData`**. This function accepts an array that is to be filled with integer values entered by the user.

b) Function **`compare`**. This function accepts two arrays. It will compare the content of both arrays for equality element-by-element. For example, if the first element of both arrays are equal, the function will display the message **"Equal"**, otherwise the message **"Not Equal"** will be displayed.

c) Function **`compute`**. This function accepts two arrays. It also declares another array to store the multiplication of each element for both arrays sent to this function. For example, if the sent arrays are **`arr1`** and **`arr2`**, and the new declared array is **`arr3`**, then **`arr3[0]`** will store the result of **`arr1[0]`** × **`arr2[0]`**. The function will return the sum of all the elements of **`arr3`**.

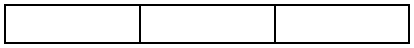
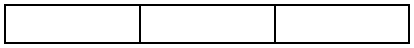
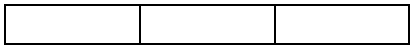
Q#5

Trace the following program segment. Write the elements of each array after execution of the statement(s) in **Table 2**.

```
1   int subs[3]={1,2,3}, mask2D[3][3]={{7},2,4,12,9,5};
2   int i, j, x = 1;
3   for(i = 0; i < 3; i++) {
4     for(j = 0; j < 2; j++)
5       mask2D[i][j] = mask2D[i][j] + subs[j];
6     subs[i] = mask2D[2-i][i];
7   }
8
9   subs[2] = subs[x++];
10  subs[1] = ++mask2D[x][x-1];
11  subs[0] = subs[x]++ - mask2D[1][0];
```

Table 2

Line	Array `subs`	Array `mask2D`
1	1 | 2 | 3	
3-7		

9		
10		
11		

Q#6

Given an incomplete program below, complete the program by filling in the blanks with an appropriate C++ statement according to the instructions stated in the comments. This program copies all the elements of a 1D array to a 2D array. Then, find the sum of square of all elements.

```
1   #include<iostream>
2   #include<cmath>
3   using namespace std;
4
5   //a) Write sumsquare function prototype
6   _____________________________________________
7
8   int main()
9   {
10    //b) Declare 2D array type integer, 3 rows 2 columns named val
11    _____________________________________________
12
13    /*c) Declare 1D array type integer named num with size declarator
14    6 with initial value: 1,2,3,4,5,6*/
15    _____________________________________________
16
17    int n = 0;
18
19    //d) Copy all elements of array num to val
20    _____________________ d(i) ____________________________
21        _____________________ d(ii) ____________________ {
22            _____________________ d(iii) ____________________
23          n++;
24        }
25
26    /*e) Call function sumsquare, pass array val and display the
27    returned value*/
28    _____________________________________________
29
30    return 0;
31  }
32
33  //sumsquare function definition
34  double sumsquare(int A[], int x) {
```

```
35    double sum = 0;
36
37    //f) Calculate sum of square of all elements in array
38                          f(i)
39                              f(ii)
40                                  f(iii)
41
42    return sum;
43 }
```

Q#7

a) Answer the following questions based on the **Program 6**:

```
1  //Program 6
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     int i, j;
8     int a[3][4] = {32, 21, 43, 54, 32, 12, 1, 59, 38, 80};
9
10    for (i = 0; i < 3; i++)
11    {
12       for (j = 0; j < 4; j++)
13             cout << "a[" << i << "][" << j << "] = "
14                  << a[i][j] << "\t";
15       cout << "\n";
16    }
17    return 0;
18 }
```

i) If the values of **i** and **j** as in the **Table 2**, write the output to be displayed.

Table 2: Values of **i** and **j**

i	**j**
0	1
1	2
2	0
0	3
2	1
2	3

ii) What is total number of values for the second column?

iii) What is total number of values for the last row?

b) Based on the **Program 7**, trace the value of the variables and determine the output of the program.

```
1   //Program 7
2   #include <iostream>
3   #include <cmath>
4   using namespace std;
5
6   int main()
7   {
8      int numbers[3][2];
9
10     for (int i = 0; i < 3; ++i)
11        for (int j = 0; j < 2; ++j)
12           numbers[i][j] = pow((i + j), 2.0);
13
14     for (int i = 0; i < 3; ++i)
15     {
16        for (int j = 0; j < 2; ++j)
17           cout << numbers[i][j] * 5 << "\t";
18        cout << endl;
19     }
20
21     return 0;
22  }
```

c) Based on the comments given (**bold** text) in the **Program 8**, complete the program with appropriate C++ statements. ***Note:*** You may need to write more than one C++ statements to answer some of the questions given in the comments.

```
1   //Program 8
2   #include <iostream>
3   using namespace std;
4
5   int main()
6   {
7       int values[5];//Array of five integers
8       int count;          //Loop counter
9       int smallest; //To hold the smallest value
10
11      cout << "\nThis program will ask you to enter five "
12           << "\nvalues, then it will determine the smallest "
13           << "\nof the values you entered." << endl;
14
15      //(i) Get five integer values from the user
16      //using loop
17      ____________________________________________________________
18
19      //(ii) Find the smallest value
20      ____________________________________________________________
21
22      //(iii) Display the results
23      ____________________________________________________________
24
25      return 0;
26  }
```

Q#8

The code segment below is meant to print the element of arrays and has four syntax and/or logical errors. Identify which line and describe each of the errors.

Line	C++ Statement
1	`int numbers[10] = {1,4, ,6, ,8};`
2	`int table[10];`
3	`char greeting[] = 'H','e','l','l','o';`
4	
5	`for (int x = 0; x < 15; x++)`
6	`{`
7	`   cout << "Enter the next value: ";`
8	`   cin  >> table [x];`
9	`}`
10	`numbers[] = table [];`
11	
12	`for (int x = 0; x < 10; x++)`
13	`    cout << numbers[x] << endl;`

Q#9

Write the output of the code segment below.

```
1   const int ROW = 5, COL = 2;
2   int i, j, arr[ROW][COL];
3   int sum[ROW] = {0, 0, 0, 0, 0};
4
5   for (i = 0; i < ROW; i++)
6   {
7         for (j = 0; j < COL; j++)
8         {
9               arr[i][j] = pow(COL, i) + j;
10              sum[i] += arr[i][j];
11        }
12        cout << sum[i] << endl;
13  }
14
15  for (i = 0; i < ROW; i++)
16        cout << sum[i] - arr[i][COL-1] << endl;
```

Q#10

Program 6 is a program to rotate right element of array that receives size of array and the integer as element in the array as input from the user. Complete a function called `rotateRight` in **Program 6** that accepts two parameters, an integer array and the size of the array. The function will move one element to the right of array except for the final element of the array which will be moved to the first element. **Figure 3** shows a sample of array, before

and after rotation of one element is applied. Meanwhile, **Figure 4** shows the example of program's input and output. (Hint: dynamic memory allocation is not required).

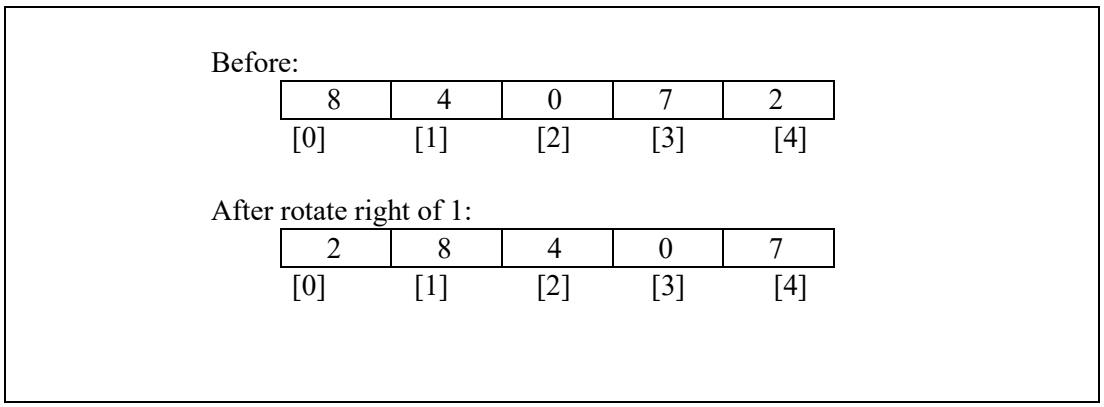


Figure 3: Sample array of rotate right for one element moves

```
Size of array: 5 [Enter]
A[0]: 8 [Enter]
A[1]: 4 [Enter]
A[2]: 0 [Enter]
A[3]: 7 [Enter]
A[4]: 2 [Enter]

Before rotation: 8 4 0 7 2
After rotation : 2 8 4 0 7
```

Figure 4: Sample of run – example of program's input and output

```
1  // Program 6
2  #include <iostream>
3  #define N 12
4  using namespace std;
5
6  void rotateRight (int A[], int size)
7  {
8  //Complete your code here
9
10
11
12
13
14
15
16
17
18
19
20
21
22 }
23
24 int main()
25 {
26       int A[N], size;
27
```

```
28          cout << "Size of array: ";
29          cin >> size;
30
31          for (int i = 0; i < size; i++)
32          {
33                cout << "A[" << i << "]: ";
34                cin >> A[i];
35          }
36
37          cout << endl;
38
39          cout << "Before rotation: ";
40
41          for (int i = 0; i < size; i++)
42                cout << A[i] << "\t";
43          cout << endl;
44
45          rotateRight(A, size);
46
47          cout << "After rotation: ";
48
49         for (int i = 0; i < size; i++)
50                cout << A[i] << "\t";
51          cout << endl;
52
53          return 0;
54    }
```