PN_CODING_CHALLENGE_2—06/06/2020

1. Write a C++ Program to print right angled pyramid of numbers.

Solution:

```
#include <iostream>
using namespace std;
int main()
{
	int i, rows, j, k=1;
	cout<<"Enter the number of rows: ";
	cin>>rows;
	for(i=1; i<=rows; i++)
	{
		for(j=1; j<=i; j++)
		{
			cout<<k<<"\t";
			k++;
		}
		cout<<"\n";
	}
	return 0;
}
```

****

2. Write a C++ Program to Multiply Two Matrix Using Multi-dimensional Arrays.

Solution:

```
#include<iostream>
using namespace std;
int main()
{
```

```
int product[10][10], r1=2, c1=3, r2=3, c2=3, i,  j, k;
int a[2][3] = { {2, 4, 1} , {2, 3, 9} };
int b[3][3] = { {1, 2, 3} , {3, 6, 1} , {2, 9, 7} };
if (c1 != r2)
{
	cout<<"Column of first matrix should be equal to row of second matrix";
}
else
{
	cout<<"The first matrix is:"<<endl;
	for(i=0; i<r1; ++i)
	{
		for(j=0; j<c1; ++j)
		cout<<a[i][j]<<" ";
		cout<<endl;
	}
	cout<<endl;
	cout<<"The second matrix is:"<<endl;
	for(i=0; i<r2; ++i)
	{
		for(j=0; j<c2; ++j)
		cout<<b[i][j]<<" ";
		cout<<endl;
	}
	cout<<endl;
	for(i=0; i<r1; ++i)
	for(j=0; j<c2; ++j)
	{
		product[i][j] = 0;
	}
	for(i=0; i<r1; ++i)
	for(j=0; j<c2; ++j)
	for(k=0; k<c1; ++k)
	{
		product[i][j]+=a[i][k]*b[k][j];
	}
	cout<<"Product of the two matrices is:"<<endl;
	for(i=0; i<r1; ++i)
	{
		for(j=0; j<c2; ++j)
		cout<<product[i][j]<<" ";
		cout<<endl;
	}
}
```

```
    return 0;
}
```

---

3. Write A C++ Program To Implement Queue Operations Using Switch Statement.

Solution:

```
#include <iostream>

using namespace std;

int queue[100], n = 100, front = - 1, rear = - 1;

void Insert()

{

    int val;

    if (rear == n - 1)

    cout<<"Queue Overflow"<<endl;

    else

    {

        if (front == - 1)

        front = 0;

        cout<<"Insert the element in queue : "<<endl;

        cin>>val;

        rear++;

        queue[rear] = val;

    }

}

void Delete()

{

    if (front == - 1 || front > rear)

    {

        cout<<"Queue Underflow ";
```

```
            return ;
        }
        else
        {
            cout<<"Element deleted from queue is : "<< queue[front] <<endl;
            front++;;
        }
}
void Display()
{
        if (front == - 1)
        cout<<"Queue is empty"<<endl;
        else
        {
            cout<<"Queue elements are : ";
            for (int i = front; i <= rear; i++)
            cout<<queue[i]<<" ";
            cout<<endl;
        }
}
int main()
{
        int ch;
        cout<<"1) Insert element to queue"<<endl;
        cout<<"2) Delete element from queue"<<endl;
        cout<<"3) Display all the elements of queue"<<endl;
        cout<<"4) Exit"<<endl;
        do
        {
```

```
		cout<<"Enter your choice : "<<endl;
		cin<<ch;
		switch (ch)
		{
			case 1: Insert();
				break;
			case 2: Delete();
				break;
			case 3: Display();
				break;
			case 4: cout<<"Exit"<<endl;
				break;
			default: cout<<"Invalid choice"<<endl;
		}
	}
	while(ch!=4);
	return 0;
}
```

****

4. Write a C++ Program For Selection Sort

Solution:

```
#include<iostream>
using namespace std;
void swapping(int &a, int &b)
{                                       //swap the content of a and b
	int temp;
	temp = a;
	a = b;
	b = temp;
```

```
}
void display(int *array, int size)
{
	for(int i = 0; i<size; i++)
	cout << array[i] << " ";
	cout << endl;
}
void selectionSort(int *array, int size)
{
	int i, j, imin;
	for(i = 0; i<size-1; i++)
	{
		imin = i;   //get index of minimum data
		for(j = i+1; j<size; j++)
		if(array[j] < array[imin])
		imin = j;
		//placing in correct position
		swap(array[i], array[imin]);
	}
}
int main()
{
	int n;
	cout << "Enter the number of elements: ";
	cin >> n;
	int arr[n];		//create an array with given number of elements
	cout << "Enter elements:" << endl;
	for(int i = 0; i<n; i++)
	{
```

```
            cin >> arr[i];
        }
        cout << "Array before Sorting: ";
        display(arr, n);
        selectionSort(arr, n);
        cout << "Array after Sorting: ";
        display(arr, n);
}
```

****

5. C++ Program to Find All Roots of a Quadratic Equation.

Solution:

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
        float a, b, c, x1, x2, discriminant, realPart, imaginaryPart;
        cout << "Enter coefficients a, b and c: ";
        cin >> a >> b >> c;
        discriminant = b*b - 4*a*c;
        if (discriminant > 0)
        {
                x1 = (-b + sqrt(discriminant)) / (2*a);
                x2 = (-b - sqrt(discriminant)) / (2*a);
                cout << "Roots are real and different." << endl;
                cout << "x1 = " << x1 << endl;
                cout << "x2 = " << x2 << endl;
        }
        else if (discriminant == 0)
        {
                cout << "Roots are real and same." << endl;
                x1 = (-b + sqrt(discriminant)) / (2*a);
```

```
                cout << "x1 = x2 =" << x1 << endl;
        }
        else
        {
                realPart = -b/(2*a);
                imaginaryPart =sqrt(-discriminant)/(2*a);
                cout << "Roots are complex and different."  << endl;
                cout << "x1 = " << realPart << "+" << imaginaryPart << "i" << endl;
                cout << "x2 = " << realPart << "-" << imaginaryPart << "i" << endl;
        }
   return 0;
}
```

****