Name: Brijesh Rameshbhai Rohit
Admission number: U19CS009

PPL-PRACTICE-PROGRAMS

Q1. Write a program that requests the user to enter two integers. The program should then calculate and report the sum of all the integers between and including the two integers.
At this point, assume that the smaller integer is entered first. For example, if the user enters 2 and 9, the program should report that the sum of all the integers from 2 through 9 is 44.

Code :

```cpp
#include <iostream>
using namespace std;

int main()
{
    int a, b;
    cout << "Enter 2 integers : ";
    cin >> a >> b;
    int sum = b * (b + 1) / 2 - a * (a - 1) / 2;
    cout << "Sum : " << sum << "\n";
    return 0;
}
```

Output :

```
PS C:\Users\brijesh\Documents\ppl\practice programs> g++ q1.cpp -o q1
PS C:\Users\brijesh\Documents\ppl\practice programs> ./q1
Enter 2 integers : 1 30
Sum : 465
PS C:\Users\brijesh\Documents\ppl\practice programs>
```

Q2. Write a program that opens a text file, reads it character-by-character to the end of the file, and reports the number of characters in the file.

Code :

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ifstream in("q1.cpp");
    char c;
    int count = 0;
    while (c = in.get())
    {
        if (c == -1)
            break;
        count++;
    }
    cout << "Number of characters: " << count << "\n";
    return 0;
}
```

Output :

```
PS C:\Users\brijesh\Documents\ppl\practice programs> g++ q2.cpp -o q2
PS C:\Users\brijesh\Documents\ppl\practice programs> ./q2
Number of characters: 224
PS C:\Users\brijesh\Documents\ppl\practice programs>
```

Q3. Write a program that reads up to 10 donation values into an array of double. (Or, if you prefer, use an array template object.) The program should terminate input on non-numeric input. It should report the average of the numbers and also report how many numbers in the array are larger than the average.

Code :

```
#include <iostream>
using namespace std;
int main()
{
    double arr[10];
    double avg = 0;
    int index = 0;
    int count = 0;
    while (index < 10 && (cin >> arr[index++]))
    {
        avg += arr[index - 1];
    }
    if (index == 10)
        index++;
    avg /= index - 1;
    for (int i = 0; i < index; i++)
    {
        count += avg < arr[i];
    }
    cout << "Average is: " << avg;
    cout << " and the number of elements greater than average is
" << count
         << endl;
    return 0;
}
```

Output :

```
PS C:\Users\brijesh\Documents\ppl\practice programs> g++ q3.cpp -o q3
PS C:\Users\brijesh\Documents\ppl\practice programs> ./q3
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Average is: 1 and the number of elements greater than average is 0
PS C:\Users\brijesh\Documents\ppl\practice programs> ./q3
1 2 3 1 2 3 1 2 3 1 2 3
Average is: 1.9 and the number of elements greater than average is 6
PS C:\Users\brijesh\Documents\ppl\practice programs>
```

Q5. Write a program that uses the following functions: Fill_array() takes as arguments the name of an array of double values and an array size.
It prompts the user to enter double values to be entered in the array. It ceases taking input when the array is full or when the user enters non-numeric input,and it returns the actual number of entries.
Show_array() takes as arguments the name of an array of double values and an array size and displays the contents of the array.
Reverse_array() takes as arguments the name of an array of double values and an array size and reverses the order of the values stored in the array.The program should use these functions to fill an array, show the array, reverse the array, show the array, reverse all but the first and last elements of the array,and then show the array.

Code :

```cpp
#include <iostream>
using namespace std;

int Fill_array(double arr[], int size)
{
    int i = 0;
    cout << "Enter double values to be entered in the array\n";
    while (i < size && (cin >> arr[i++]))
        ;
    return i - 1;
}
```

```
void Show_array(double arr[], int size)
{
    for (int i = 0; i < size; i++)
        cout << arr[i] << " \n"[i == size - 1];
}

void Reverse_array(double arr[], int size)
{
    for (int i = 0; i < size - i - 1; i++)
        swap(arr[i], arr[size - i - 1]);
}

int main()
{
    int size;
    cout << "Enter the maxsize of array\n";
    cin >> size;
    double arr[size];
    size = Fill_array(arr, size);
    cout << "The array entered:\n";
    Show_array(arr, size);
    Reverse_array(arr, size);
    cout << "After reversing the array\n";
    Show_array(arr, size);
    if (size > 2)
        Reverse_array(arr + 1, size - 2);
    cout << "After reversing again accept first and last
elements\n";
    Show_array(arr, size);
    return 0;
}
```

Output :

```
PS C:\Users\brijesh\Documents\ppl\practice programs> g++ q5.cpp -o q5
PS C:\Users\brijesh\Documents\ppl\practice programs> ./q5
Enter the maxsize of array
10
Enter double values to be entered in the array
1 2 1 2 3 4 2 6
s
The array entered:
1 2 1 2 3 4 2 6
After reversing the array
6 2 4 3 2 1 2 1
After reversing again accept first and last elements
6 2 1 2 3 4 2 1
```

Q16. The CandyBar structure contains three members. The first member holds the brand name of a candy bar. The second member holds the weight (which may have a fractional part) of the candy bar, and the third member holds the number of calories (an integer value) in the candy bar. Write a program that declares such a structure and creates a CandyBar variable called snack, initializing its members to "Mocha Munch", 2.3, and 350, respectively. The initialization should be part of the declaration for snack. Finally, the program should display the contents of the snack variable.

Code :

```cpp
#include <iostream>
using namespace std;

struct CandyBar
{
    string brand_name;
    float weight;
    int calories;
};

int main()
```

```
{
    CandyBar snack = {"Mocha Munch", 2.3, 350};
    cout << "Contents of snack variable:\n";
    cout << "snack.brand_name: " << snack.brand_name << "\n";
    cout << "snack.weight: " << snack.weight << "\n";
    cout << "snack.calories: " << snack.calories << "\n";
    return 0;
}
```

Output :

```
PS C:\Users\brijesh\Documents\ppl\practice programs> g++ q16.cpp -o q16
PS C:\Users\brijesh\Documents\ppl\practice programs> ./q16
Contents of snack variable:
snack.brand_name: Mocha Munch
snack.weight: 2.3
snack.calories: 350
PS C:\Users\brijesh\Documents\ppl\practice programs>
```

Q17. Write a program that uses an array of char and a loop to read one word at a time until the word done is entered.The program should then report the number of words entered (not counting done). A sample run could look like this:
Enter words (to stop, type the word done):
anteater birthday category dumpster
envy finagle geometry done for sure
You entered a total of 7 words.
You should include the cstring header file and use the strcmp() function to make the comparison test.

Code :

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
```

```
{
    cout << "Enter words (to stop, type the word done):\n";
    string word;
    int count = 0;
    do
    {
        cin >> word;
        count++;
    } while (strcmp(word.c_str(), "done"));
    cout << "You entered a total of " << count - 1 << "
words.\n";
    return 0;
}
```

Output :

```
PS C:\Users\brijesh\Documents\ppl\practice programs> g++ q17.cpp -o q17
PS C:\Users\brijesh\Documents\ppl\practice programs> ./q17
Enter words (to stop, type the word done):
Hello there,my name is Brijesh
And I'm currently in Btech-III
And I've done multiple web dev projects.
You entered a total of 12 words.
PS C:\Users\brijesh\Documents\ppl\practice programs>
```

Q22. The CandyBar structure contains three members. The first member holds the brand name of a candy bar. The second member holds the weight (which may have a fractional part) of the candy bar, and the third member holds the number of calories (an integer value) in the candy bar. Write a program that uses a function that takes as arguments a reference to CandyBar, a pointer-to-char, a double, and an int and uses the last three values to set the corresponding members of the structure. The last three arguments should have default values of "Millennium Munch," 2.85, and 350. Also, the program should use a function that takes a reference to a CandyBar as an argument and displays the contents of the structure. Use const where appropriate.

Code :

```
#include <iostream>
using namespace std;
struct CandyBar
{
    string brand_name;
    double weight;
    int calories;
};
void init(CandyBar &candyBar,
          const char *brand_name = "Millenium Munch",
          const double weight = 2.85,
          const int calories = 350)
{
    candyBar = (CandyBar){brand_name, weight, calories};
}
void print(const CandyBar &candyBar)
{
    cout << "Brand name: " << candyBar.brand_name << "\n";
    cout << "Weight: " << candyBar.weight << "\n";
    cout << "Calories: " << candyBar.calories << "\n";
}
int main()
```

```
{
    CandyBar candyBar;
    init(candyBar);
    print(candyBar);
    return 0;
}
```

Output :

```
PS C:\Users\brijesh\Documents\ppl\practice programs> g++ q22.cpp -o q22
PS C:\Users\brijesh\Documents\ppl\practice programs> ./q22
Brand name: Millenium Munch
Weight: 2.85
Calories: 350
PS C:\Users\brijesh\Documents\ppl\practice programs>
```