

Date: 7 July 2023

Assignment 01

By - Himanshu Tiwari

Roll number - 16

MSc. CSA Sem-2



Q1.

// Write a program to add two variables x and y using call by reference.

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

void add(int &a, int &b)
{
    a += b; // Add the value of b to a
}

int main()
{
    int x = 5;
    int y = 3;

    cout << "Before addition: x = " << x << ", y = " << y << endl;

    add(x, y); // Pass x and y by reference to the add() function

    cout << "After addition: x = " << x << ", y = " << y << endl;

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

Code + - [ ] [ ] ... ^ x

```
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1> cd "c:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1\" ; if ($?) { g++ addTwoNumbersUsingCallByReference.cpp -o addTwoNumbersUsingCallByReference } ; if ($?) { .\addTwoNumbersUsingCallByReference }
Before addition: x = 5, y = 3
After addition: x = 8, y = 3
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1>
```

Q2.

// write a c++ program to find the maximum number between three number, the variables are x,y and z.

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

```
int maxFinder(int a, int b, int c)
{
    if (a >= b && a >= c)
    {
        // returning the maximum value
        return a;
    }
    else
    {
        /*if condition is false then it means that either of two values
are greater than or equal to*/
        /*the third one. So we will compare them and find out which one
has more number.*/
        if ((b - a) > (c - a))
        {
            return b; // returning second largest element as per given
conditions in problem statement
        }
        else
        {
            return c;
        }
    }
}
```

```
int main()
{
    int x, y, z;
    cout << "Enter three numbers: ";
    cin >> x >> y >> z;
    int result = maxFinder(x, y, z);
    cout << "\nThe Maximum Number among Three Numbers Entered Is :
"<<result<<endl;
```

```
    return 0;
}
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1> cd "c:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1\" ; if ($?) { g++ maxOfThreeNumber.cpp -o maxOfThreeNumber } ; if ($?) { .\maxOfThreeNumber }
Enter three numbers: 11 33 22

The Maximum Number among Three Numbers Entered Is : 33
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1> |
```

### Q3.

// WAP to show all the permutation of your first name using string function and using pointer with strings in C++.

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

int COUNT = 1;
/*Function for swapping two characters*/
void swap(char* a, char* b) {
    char temp = *a;
    *a = *b;
    *b = temp;
}

// Function to print all permutations of string
void permutations(string& a, int l, int r)
{
    if (l == r)
        cout << a << endl;
    else {
        for (int i = l; i <= r; i++) {
            swap(a[l], a[i]);
            permutations(a, l + 1, r);
            swap(a[l], a[i]);
        }
    }
}
```

```

        ++COUNT;
    }

int main()
{
    string str = "himanshu";
    int n = str.size();
    permutations(str, 0, n - 1);
    cout<<COUNT;
    return 0;
}

```

```

uhimshna
uhimshan
uhimsahn
uhimsanh
uhimhsna
uhimhsan
uhimhnsa
uhimhnas
uhimhans
uhimhasn
uhimashn
uhimasnh
uhimahsn
uhimahns
uhimanhs
uhimansh
69282
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1>

```

I printed total number of permutations of my name i.e. 69282

Q4.

// Write a C++ program to calculate the length of the string using pointer and concatenate by string function.

```

#include <iostream>
#include <string>
using namespace std;

int strLength(const char *str)
{
    int length = 0;
    while (*str != '\0')
    {

```

```

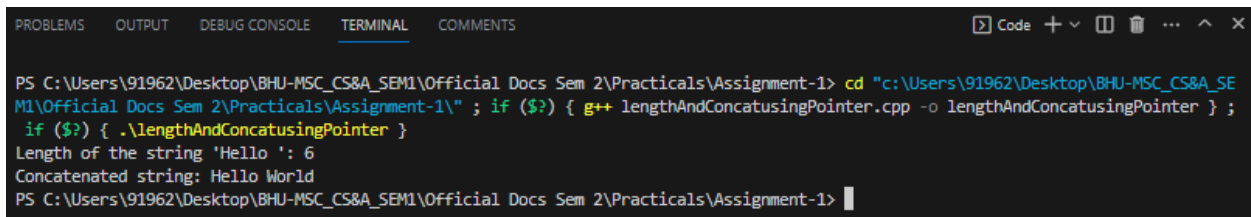
        length++;
        str++;
    }
    return length;
}

int main()
{
    const char *str = "Hello ";
    const char *name = "World";

    int length = strLength(str);
    cout << "Length of the string '" << str << "': " << length << endl;

    char result[50];
    strcpy(result, str);
    strcat(result, name);
    cout << "Concatenated string: " << result << endl;
    return 0;
}

```



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1> cd "c:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1\" ; if ($?) { g++ lengthAndConcatusingPointer.cpp -o lengthAndConcatusingPointer } ; if ($?) { .\lengthAndConcatusingPointer }
Length of the string 'Hello ': 6
Concatenated string: Hello World
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1>

```

5.

```

/*
Write a C++ program to sort an array using Pointers
*/

#include <iostream>
using namespace std;

void sortArray(int *arr, int size)
{
    for (int i = 0; i < size - 1; i++)
    {

```

```

        for (int j = 0; j < size - i - 1; j++)
        {
            if (*(arr + j) > *(arr + j + 1))
            {
                // Swap the elements
                int temp = *(arr + j);
                *(arr + j) = *(arr + j + 1);
                *(arr + j + 1) = temp;
            }
        }
    }
}

int main()
{
    int arr[] = {1, 3, 2, 5, 0, 7};
    int size = sizeof(arr) / sizeof(arr[0]);

    cout << "Array before sorting: ";
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;

    sortArray(arr, size);

    cout << "Array after sorting: ";
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;

    return 0;
}

```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1> cd "c:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1\" ; if ($?) { g++ sortArrayUsingPointer.cpp -o sortArrayUsingPointer } ; if ($?) { .\sortArrayUsingPointer }
Array before sorting: 1 3 2 5 0 7
Array after sorting: 0 1 2 3 5 7
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1>
```

6.

```
// C++ program to illustrate simple call by value and call by reference

#include <iostream>
using namespace std;

/* Function definition for addition using Call By Value Method. Here we
pass two integers */

void add(int a, int b)
{
    a += b;
}

/* Function definition for addition using Call By Reference Method. Here
we pass two integers */

void addRef(int *p_num1, int *p_num2)
{
    (*p_num1) += (*p_num2);
}

int main()
{
    // call by value
    int num = 50;
    cout << "Before calling the function: " << num << endl;
    add(num, 30);
    cout << "After Calling The Addition Using CallByValue :" << num <<
endl;

    // Call by Reference Example
    int x = 789465;
    int y = -1234;
    printf("Before adding values of %d , %d\n",x ,y );
```

```

    addRef(&x,&y);    /* Passing address as arguments*/
    printf("\n After Adding Values %d , %d",x,y);
    return 0;
}

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  COMMENTS
Code + -  [ ] [ ] ... ^ X

PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1> cd "c:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1\" ; if ($?) { g++ callByValueAndCallByRef.cpp -o callByValueAndCallByRef } ; if ($?) { .\callByValueAndCallByRef }
Before calling the function: 50
After Calling The Addition Using CallByValue :50
Before adding values of 789465 , -1234

After Adding Values 788231 , -1234
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1>

```

7.

```

// Write a program to find the size of different pointer types in C++

#include <iostream>
using namespace std;

int main()
{
    cout << "Size of char: " << sizeof(char) << endl;
    // size of character is
    cout << "Size of short:" << sizeof(short) << endl;
    // size of short
    cout << "Size of long :" << sizeof(long) << endl;
    // size of long
    cout << "Size of float :" << sizeof(float) << endl;
    // size of float
    cout << "Size of double : " << sizeof(double) << endl;
    // size of
    cout << "Size of void* : " << sizeof(void *) << endl;
    // size
    cout << "Size of Integer Pointer : " << sizeof(int *) << endl;
    // pointer
    cout << "Size of Char Pointer : " << sizeof(char *) << endl;
    // char

```



```
    cout << "Size of Double Pointer : " << sizeof(double *) << " bytes" <<
endl; // pointer

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

Code + - [ ] [ ] ... ^ X

```
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1> cd "c:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1\" ; if ($?) { g++ sizeOfDifferentPointerTypes.cpp -o sizeOfDifferentPointerTypes } ;
if ($?) { .\sizeOfDifferentPointerTypes }
Size of char: 1
Size of short:2
Size of long :4
Size of float :4
Size of double : 8
Size of void* : 8
Size of Integer Pointer : 8
Size of Char Pointer : 8
Size of Double Pointer : 8 bytes
PS C:\Users\91962\Desktop\BHU-MSC_CS&A_SEM1\Official Docs Sem 2\Practicals\Assignment-1>
```