Programming Fundamentals 8-

- Program 1:

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

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

int sumOFdigits (int number)

    sum = 0;
    while (number != 0)
{
    sum += number % 10;
    number /= 10;
}
    return sum;

int main ( )
{
    int number;
    cout << "Enter a number \n";
    cin >> number;
```

```
int sum= digitOfnumbers(number);
cout<<"Sum of digits is "<<sum;
return 0;
}
```

Output:

Enter a number
415
Sum of digits is 10

Explanation:

- We will ask the user to enter a number.
- Declare another variable sum where sum will be stored.
- Using a while loop, we will extract the values and get the sum.

Program 2:

```
#include <iostream>
using namespace std;
int reverse (int number)
{
    int reversed = 0;
    while (number > 0)
```

```
int sum= digitOfnumbers(number);
cout<<"Sum of digits is "<<sum;
return 0;
}
```

Output:

Enter a number
415
Sum of digits is 10

Explanation:

- We will ask the user to enter a number.
- Declare another variable sum where sum will be stored.
- Using a while loop, we will extract the values and get the sum.

Program 2:

```
#include <iostream>
using namespace std;
int reverse (int number)
{
    int reversed = 0;
    while (number > 0)
```

```
{
    reversed *= 10 + number % 10;
    number /= 10;
}
    return reversed;
}

bool isPalindrome(int number)
{
    return number == reverse(number);
}

int main()
{
    int number;
    cout << "Enter a number\n";
    cin >> number;

    if (isPalindrome(number))
{
    cout << "The number is palindrome\n"
}
    else
    {
    cout << "The number is not palindrome.\n";
    }
return 0;
```

Output :-

Enter a number
122
The number is not palindrome.

Explanation :-

- The reverse function will reverse the integer entered by the user.
- The isPalindrome function checks if the number reversed is same as before or not.
- If it is same then display output accordingly.

Program 3:

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

void SortedNumbers (float n1, int n2, int n3)
{
    float temp;
    if (n1 > n2)
    {
        temp = num1;
        num1 = num2;
        num2 = num3;
    }
```

```
if (num2 > num3)
{
    temp = num2;
    num2 = num3;
    num3 = temp;
}

if (num3 > num1)
{
    temp = num1;
    num1 = num2;
    num2 = temp;
}
 cout<< "The sorted numbers are \n";
 cout << num1 << "," << num2 << ","
        << num3;
}

int main() {
 float num1, num2, num3;
 Cout<< "Enter 3 numbers \n";
 cin >> num1 >> num2 >> num3;

 SortedNumbers (num1, num2, num3);
 return 0;

}
```

Output:

Enter three numbers
2 8 4
The sorted numbers are 2 4 8
7, 3, 4

Explanation:-

- Ask the user to enter three numbers.
- The sortedNumbers function will be used to sort the three numbers in ascending order. The function will compare the three numbers and swaps them. This process continues until numbers are sorted.

Program 4:

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

string convertmillis(intmillis)
{
    int hours, minutes, secs;
    hours = millis/3600000;
    millis %= 3600000;
    minutes = millis/60000;
    millis %= 60000;
    secs = millis/1000;
```

```
    return to_string(hours) + to_string
        (minutes) + to_string(seconds);
}
int main()
{
    int millis;
    cout<<"Enter a value for millisec\n";
    cin>>millis;

    cout<<" New Time : \n" ;
    cout<<convertMillis(millis);

    return 0;
}
```

Output:

```
Enter a value for milliseconds:555550000
New Time :
154:19:10.
```

Program 5:

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

void TowerOfHanoi(int numdisks, int source,
        int destination, int temp);
```

```
int main() {
  int numDisks;
  cout << "Enter the no. of disks \n";
  cin >> numdisks;

  cout << " Steps to solve Tower of Hanoi"
    << numdisks << "disks";

  TowerOfHanoi (numdisks 1,3,2);

  return 0;
}

void TowerOfHanoi (int numdisks, int source,
      int destination, int temp)

{
  if (numdisks == 1) {
    cout << source << " -> " destination;
    return;
  }

  TowerOfHanoi (numdisks - 1, source, temp, destinati);
cout << source << " -> " << destination << endl;
 TowerOfHanoi (numdisks - 1, temp, source, destination);
}
```

DATE: __/__/____

Output:-

Enter the no. of disks :
2
Steps to solve tower of Hanoi with 2 disks.

1 → 2

1 → 3

2 → 1