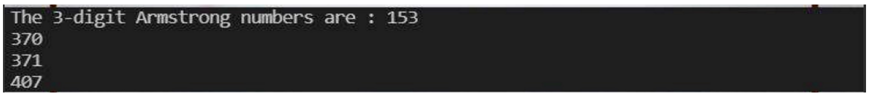
LAB - 3

Q1. Wap to find all armstrong no from 100 to 999 by using function.

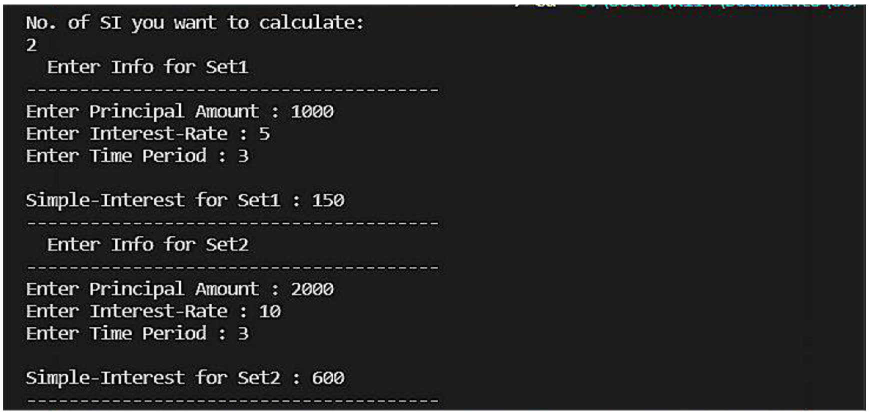
```
#include <iostream>
using namespace std;
int main(void)
{
  int i = 100, rmdr, sum, temp, tnum;
  cout << "The 3-digit Armstrong numbers are :";
  for (i = 100; i <= 999; i++)
  {
    tnum = i;
    sum = 0;
    while (tnum > 0)
    {
      rmdr = tnum % 10;
      temp = rmdr * rmdr * rmdr;
      sum = temp + sum;
      tnum = tnum / 10;
    }
    if (i == sum)
      cout << sum << endl;
  }
}
```



```
The 3-digit Armstrong numbers are : 153
370
371
407
```

Q2. WAP to display the simple interest by using function given p, t, r.

```
#include <iostream>
using namespace std;
float interest(int P, float R, int N)
{
  float SI;
  SI = P * R * N / 100.0;
  return SI;
}
int main()
{
  int p, n, i, nsi;
  float r, Z;
  cout << "No. of SI you want to calculate:" << endl;
  cin >> nsi;
  for (i = 1; i <= nsi; i++)
  {
    cout << " Enter Info for Set" << i;
    cout << "\n--------------------------------------\n";
    cout << "Enter Principal Amount :";
    cin >> p;
    cout << "Enter Interest-Rate :";
    cin >> r;
    cout << "Enter Time Period :";
    cin >> n;
    Z = interest(p, r, n);
    cout << "\nSimple-Interest for Set" << i << ":" << Z << endl;
    cout << "--------------------------------------\n";
  }
  return 0;
}
```

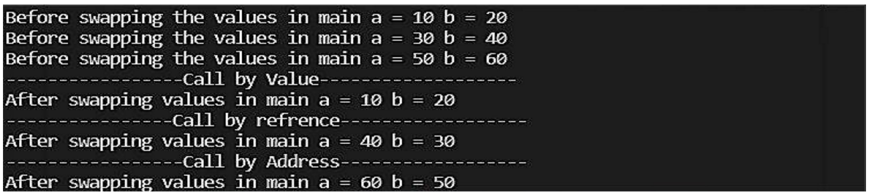


```
No. of SI you want to calculate:
2
  Enter Info for Set1
---------------------------------------
Enter Principal Amount : 1000
Enter Interest-Rate : 5
Enter Time Period : 3

Simple-Interest for Set1 : 150
---------------------------------------
  Enter Info for Set2
---------------------------------------
Enter Principal Amount : 2000
Enter Interest-Rate : 10
Enter Time Period : 3

Simple-Interest for Set2 : 600
---------------------------------------
```

Q3. Wap to swap 2 no by using call by value, call by address and call by reference.

```
#include <iostream>
using namespace std;
void swap(int a, int b)
{
  int temp;
  temp = a;
  a = b;
  b = temp;
}
void swapref(int &c, int &d)
{
  int temp;
  temp = c;
  c = d;
  d = temp;
}
void swapAdd(int *m, int *n)
{
  int temp = *m;
  *m = *n;
  *n = temp;
}
int main()
{
  int a = 10;
  int b = 20;
  int c = 30;
  int d = 40;
  int e = 50;
  int f = 60;
  cout << "Before swapping the values in main a = " << a << " b = " << b << endl;
  cout << "Before swapping the values in main a = " << c << " b = " << d << endl;
  cout << "Before swapping the values in main a = " << e << " b = " << f << endl;
  cout << "---------------Call by Value----------------" << endl;
  swap(a, b);
```

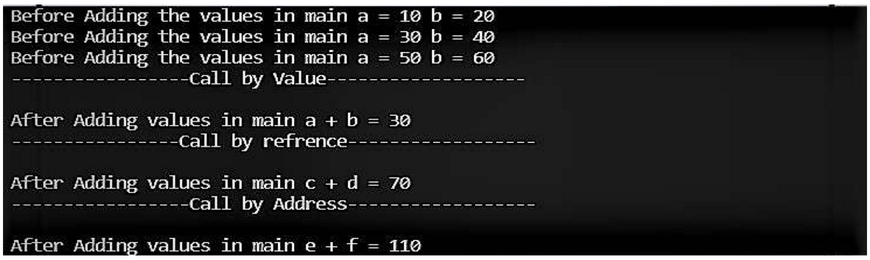
```
    cout << "After swapping values in main a = " << a << " b = " << b << endl;
    cout << "---------------Call by refrence----------------" << endl;
    swapref(c, d);
    cout << "After swapping values in main a = " << c << " b = " << d << endl;
    cout << "----------------Call by Address----------------" << endl;
    swapAdd(&e, &f);
    cout << "After swapping values in main a = " << e << " b = " << f << endl;
}
```



```
Before swapping the values in main a = 10 b = 20
Before swapping the values in main a = 30 b = 40
Before swapping the values in main a = 50 b = 60
----------------Call by Value-----------------
After swapping values in main a = 10 b = 20
---------------Call by refrence----------------
After swapping values in main a = 40 b = 30
----------------Call by Address----------------
After swapping values in main a = 60 b = 50
```

Q4. Wap to add two no by using call by value, call by reference and call by address.

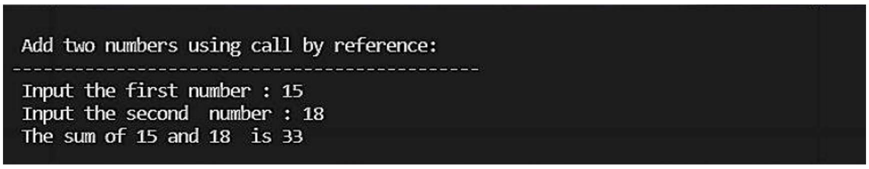
```
#include <iostream>
using namespace std;
void Add(int a, int b)
{
    cout << endl;
}
void Addref(int &c, int &d)
{
   cout << endl;
}
void AddAdd(int *m, int *n)
{
    cout <<endl ;
}
int main()
{
    int a = 10;
    int b = 20;
    int c = 30;
    int d = 40;
    int e = 50;
    int f = 60;
    cout << "Before Adding the values in main a = " << a << " b = " << b << endl;
    cout << "Before Adding the values in main a = " << c << " b = " << d << endl;
    cout << "Before Adding the values in main a = " << e << " b = " << f << endl;
    cout << "----------------Call by Value-----------------" << endl;
    Add(a, b);
    cout << "After Adding values in main a + b = " << a + b << endl;
    cout << "---------------Call by refrence----------------" << endl;
    Addref(c, d);
    cout << "After Adding values in main c + d = " << c + d << endl;
    cout << "----------------Call by Address----------------" << endl;
    AddAdd(&e, &f);
    cout << "After Adding values in main e + f = " << e + f << endl;
}
```



```
Before Adding the values in main a = 10 b = 20
Before Adding the values in main a = 30 b = 40
Before Adding the values in main a = 50 b = 60
----------------Call by Value-----------------

After Adding values in main a + b = 30
---------------Call by refrence-----------------

After Adding values in main c + d = 70
----------------Call by Address-----------------

After Adding values in main e + f = 110
```

Q5 Wap to add two no by using two reference variable.

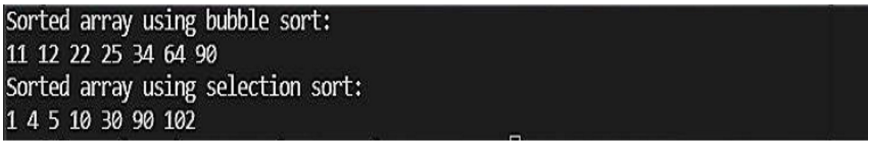
```
#include <iostream>
using namespace std;
int addTwoNumbers(int *, int *);
int main()
{
   int fno, sno, sum;
   cout << "\n\n Add two numbers using call by reference:\n";
   cout << "-------------------------------------------\n";
   cout << " Input the first number : ";
   cin >> fno;
   cout << " Input the second  number : ";
   cin >> sno;
  sum = addTwoNumbers(&fno, &sno);
  cout<<" The sum of "<< fno <<" and "<< sno <<" is "<< sum <<"\n\n";
  return 0;
}
int addTwoNumbers(int *n1, int *n2)
{
   int sum;
   sum = *n1 + *n2;
   return sum;
}
```



```
 Add two numbers using call by reference:
--------------------------------------------
 Input the first number : 15
 Input the second  number : 18
 The sum of 15 and 18  is 33
```

Q6. Wap sort the array elements ascending order in C++ by using both selection and bubble sort.

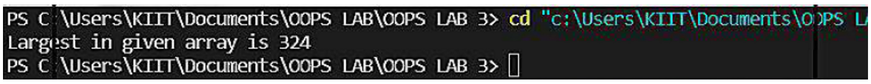
```
#include <iostream>
using namespace std;
void swap(int *xp, int *yp)
{
   int temp = *xp;
   *xp = *yp;
   *yp = temp;
}
void bubbleSort(int arr[], int n)
{
   int i, j;
   for (i = 0; i < n - 1; i++)
      for (j = 0; j < n - i - 1; j++)
         if (arr[j] > arr[j + 1])
            swap(&arr[j], &arr[j + 1]);
}
void selectionSort(int arr[], int n)
{
    int i, j, min_idx;
    for (i = 0; i < n - 1; i++)
    {
        min_idx = i;
        for (j = i + 1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;
        swap(&arr[min_idx], &arr[i]);
    }
}
void printArray(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;
}
int main()
{
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int arr2[] = {10, 30, 5, 4, 102, 1, 90};
    int n = sizeof(arr) / sizeof(arr[0]);
    bubbleSort(arr, n);
    cout << "Sorted array using bubble sort: \n";
    printArray(arr, n);
    selectionSort(arr2, n);
    cout << "Sorted array using selection sort: \n";
    printArray(arr2, n);
    return 0;
}
```



```
Sorted array using bubble sort:
11 12 22 25 34 64 90
Sorted array using selection sort:
1 4 5 10 30 90 102
```

Q7. WAP function will return max no. using arr.

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

int largest(int arr[], int n)
{
    int i;
    int max = arr[0];
    for (i = 1; i < n; i++)
        if (arr[i] > max)
            max = arr[i];
    return max;
}
int main()
{
    int arr[] = {10, 324, 45, 90, 98};
    int n = sizeof(arr) / sizeof(arr[0]);
    cout << "Largest in given array is "<< largest(arr, n);
    return 0;
}
```



```
PS C:\Users\KIIT\Documents\OOPS LAB\OOPS LAB 3> cd "c:\Users\KIIT\Documents\OOPS L
Largest in given array is 324
PS C:\Users\KIIT\Documents\OOPS LAB\OOPS LAB 3>
```