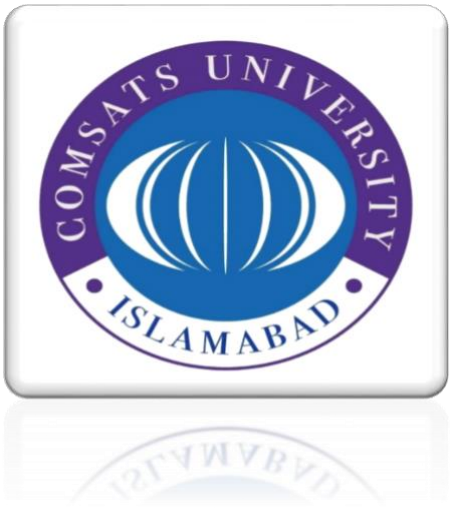
# ***Comsats University  Islamabad (Vehari campus)***




## ***DSA Lab (Assignment 1)***

***Submitted by:***

*Zainab Noor*

***Roll no:***

*SP22-BCS-082(Section B)*

***Department:***

*Computer Science*

***Subject:***

*Data structure and algorithm*

***Submitted to:***

*Mam Yasmeen Jana*

## Qno 1: How to create a GitHub account?

**Ans: Step 1: Access the GitHub Website Open your preferred web browser.**

In the address bar, enter the URL for the GitHub website: https://github.com/.

**Step 2: Sign Up**

**3**. On the GitHub homepage, locate and click the "Sign up" button in the top-right corner of the page.

**Step 3: Provide Basic Information**

**4**. You will be directed to the "Create your account" page. Here, you need to fill in the following information:

**Username**: Choose a unique username for your GitHub account.

**Email address:** Enter a valid email address.

**Password**: Create a strong, secure password for your account.

**Step 4: Click "Continue"**

**5**. After entering your information, click the green "**Continue**" button.

**Step 5: Choose a Username**

**6.** If the username you selected is already taken, GitHub will prompt you to choose an alternative username. Follow the instructions to select a unique username.

**Step 6: Verify Your Email**

**7**. **GitHub** will send a verification email to the address you provided. Open your email inbox and locate the email from **GitHub**.

Click on the verification link in the email to confirm your email address.

**Step 7: Complete Your Profile (Optional)**

**9.** You have the option to complete your GitHub profile by adding your name and a profile picture. This step is optional but can help others identify you on the platform.

**Step 8: Submit Your Information**

**10**. After completing your profile (or skipping it), click the "**Submit**" button to create your GitHub account.

**Step 9: Personalize Your Experience**

**11**. **GitHub** may prompt you to select your preferences and interests. Customize these settings to tailor your GitHub experience.

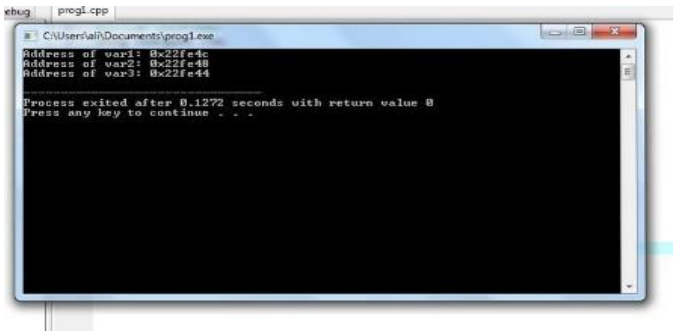
## Qno 2:

### Program 1:

Printing Variable Addresses in C++

```
#include <iostream>

Using namespace std;

Int main()

{

   // declare variables

   Int var1 = 3;

   Int var2 = 24;

   Int var3 = 17;

   // print address of var1

   Cout << “Address of var1: “<< &var1 << endl;

   // print address of var2

   Cout << “Address of var2: “ << &var2 << endl;

   // print address of var3

   Cout << “Address of var3: “ << &var3 << endl;

}
```

**Output:**

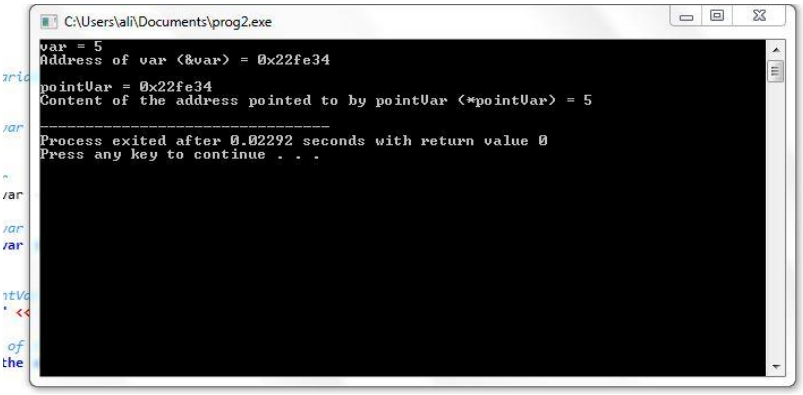


**Program 2:**

Working of C++ Pointers

```
#include <iostream>
Using namespace std;
Int main() {
   Int var = 5;
   // declare pointer variable
   Int* pointVar;
   // store address of var
   pointVar = &var;
   // print value of var
   Cout << “var = “ << var << endl;
   // print address of var
   Cout << “Address of var (&var) = “ << &var << endl
       << endl;
   // print pointer pointVar
   Cout << “pointVar = “ << pointVar << endl;
   // print the content of the address pointVar points to
   Cout << “Content of the address pointed to by pointVar (*pointVar) = “ << *pointVar << endl;
   Return 0;
}
```

**Output:**



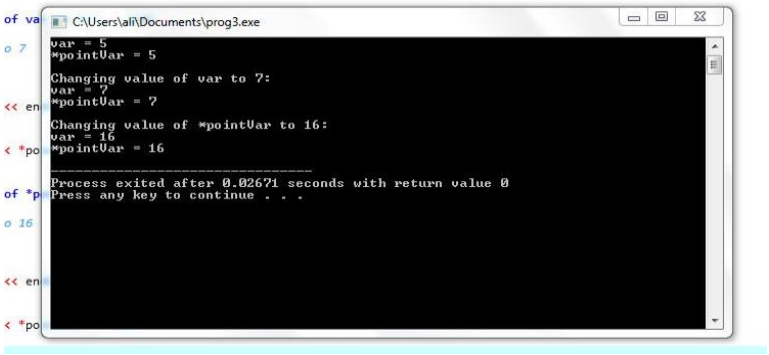

## Program 3:

Changing Value Pointed by Pointers

```
#include <iostream>
Using namespace std;
Int main() {
   Int var = 5;
   Int* pointVar;
   // store address of var
   pointVar = &var;
   // print var
   Cout << “var = “ << var << endl;
   // print *pointVar
   Cout << “*pointVar = “ << *pointVar << endl
      << endl;
   Cout << “Changing value of var to 7:” << endl;
   // change value of var to 7
   Var = 7;
   // print var
```

```
    Cout << “var = “ << var << endl;
    // print *pointVar
    Cout << “*pointVar = “ << *pointVar << endl
        << endl;
    Cout << “Changing value of *pointVar to 16:” << endl;
    // change value of var to 16
    *pointVar = 16;
    // print var
    Cout << “var = “ << var << endl;
    // print *pointVar
    Cout << “*pointVar = “ << *pointVar << endl;
    Return 0;
}
```

**Output:**




## Program 4:

```
#include <iostream>
#include <ctime>
Using namespace std;
Void getSeconds(unsigned long *par);
Int main () {
```

```
  Unsigned long sec;
  getSeconds( &sec );
  // print the actual value
  Cout << “Number of seconds :” << sec << endl;
  Return 0;
}
Void getSeconds(unsigned long *par) {
  // get the current number of seconds
  *par = time( NULL );
  Return;
}
```

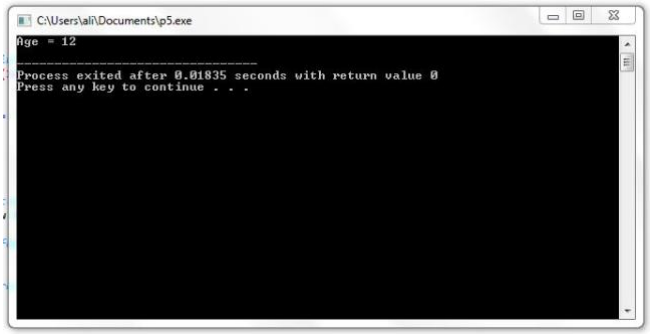
**Output:**


**Program 5:**

```
#include <iostream>
using namespace std;
class Student {
  private:
    int age;
  public:
    // constructor initializes age to 12
    Student() : age(12) {}
```

```
    void getAge() {
      cout << "Age = " << age << endl;
    }
};
int main() {
  // dynamically declare Student object
  Student* ptr = new Student();
  // call getAge() function
  ptr->getAge();
 // ptr memory is released
  delete ptr;
  return 0;
}
```

**Output:**

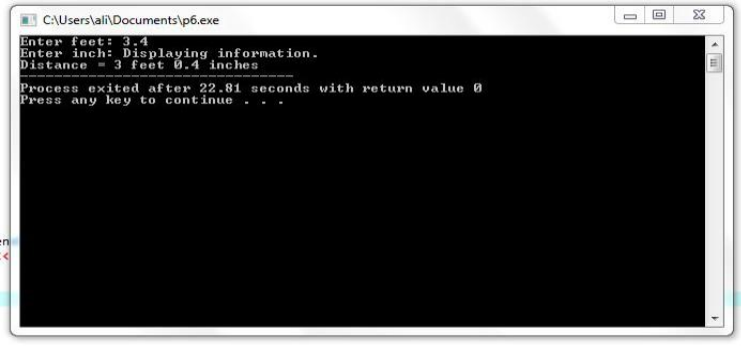



**Program 6:**

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

struct Distance {
    int feet;
    float inch;
```

```
};

int main() {
    Distance *ptr, d;

    ptr = &d;

    cout << "Enter feet: ";
    cin >> (*ptr).feet;
    cout << "Enter inch: ";
    cin >> (*ptr).inch;

    cout << "Displaying information." << endl;
    cout << "Distance = " << (*ptr).feet << " feet " << (*ptr).inch << " inches";

    return 0;
}
```

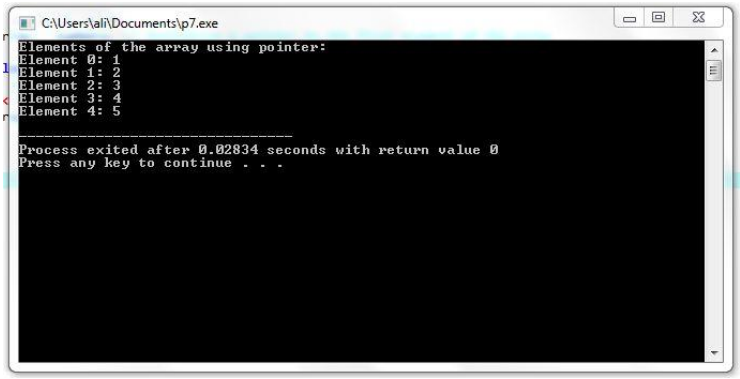
**Output:**




**Program 7:**

```
#include <iostream>
int main() {
   int numbers[] = {1, 2, 3, 4, 5};
   int* pointerToArray = numbers; // Initialize a pointer to the first element of the array

   std::cout << "Elements of the array using pointer:" << std::endl;
   for (int i = 0; i < 5; ++i) {
      std::cout << "Element " << i << ": " << *pointerToArray << std::endl;
      pointerToArray++; // Move the pointer to the next element
   }

   return 0;
}
```

## Output:

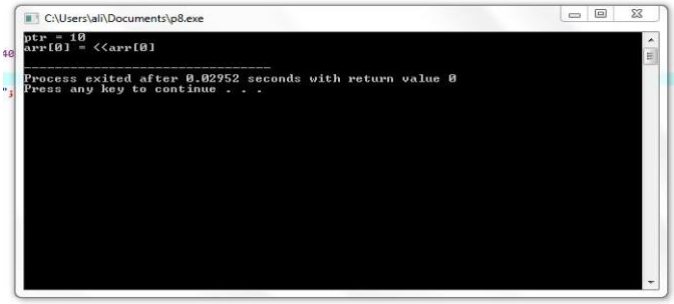



## Program 8:

```
#include<iostream>
Using namespace std;
Int main()
{
```

```
  Int *ptr;
  Int arr[5] = {10, 20, 30, 40, 50};
  Ptr = arr;
  Cout<<"ptr = "<<*ptr;
  Cout<<"\narr[0] = "<<arr[0];
  Cout<<endl;
  Return 0;
}
```

## Output:

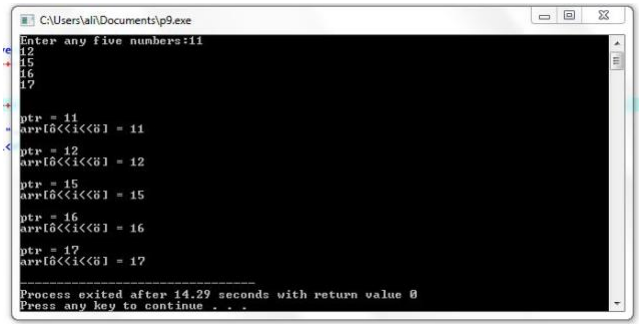



## Program 9:

```
#include<iostream>
Using namespace std;
Int main()
{
  Int *ptr, arr[5], I;
  Cout<<"Enter any five numbers: ";
  For(i=0; i<5; i++)
    Cin>>arr[i];
  Ptr = arr;
  For(i=0; i<5; i++)
```

```
  {
    Cout<<"\n\nptr = "<<*ptr;
    Cout<<"\narr["<<i<<"] = "<<arr[i];
    Ptr++;
  }
  Cout<<endl;
  Return 0;
}
```

**Output:**

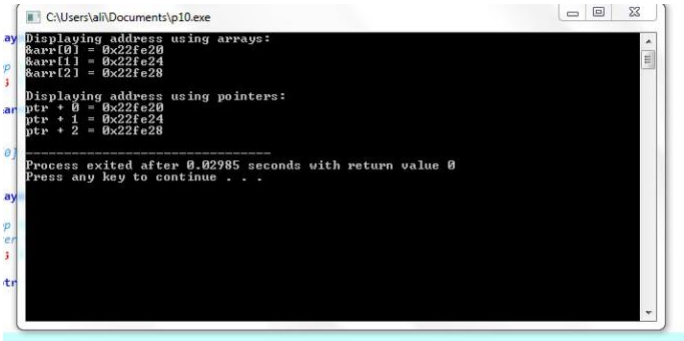



**Program 10:**

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

Int main()
{
  Char name[] = "CodesCracker";
  Char *cptr;
  Cptr = name;
  While(*cptr != '\0')
```

```
  {
    Cout<<*cptr;
    Cptr++;
  }
  Cout<<endl;
  Return 0;
}
```

**Output:**

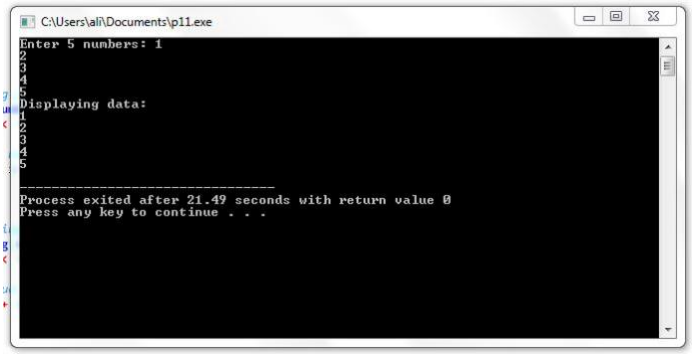



**Program 11:**

```
#include <iostream>
#include <string>
Using namespace std;
Int main() {
  String food = “Pizza”;
  String* ptr = &food;
  // Output the value of food
  Cout << food << “\n”;
  // Output the memory address of food
  Cout << &food << “\n”;
  // Access the memory address of food and output its value
```

```
 Cout << *ptr << “\n”;
 // Change the value of the pointer
 *ptr = “Hamburger”;
 // Output the new value of the pointer
 Cout << *ptr << “\n”;
 // Output the new value of the food variable
 Cout << food << “\n”;
 Return 0
}
```

**Output:**

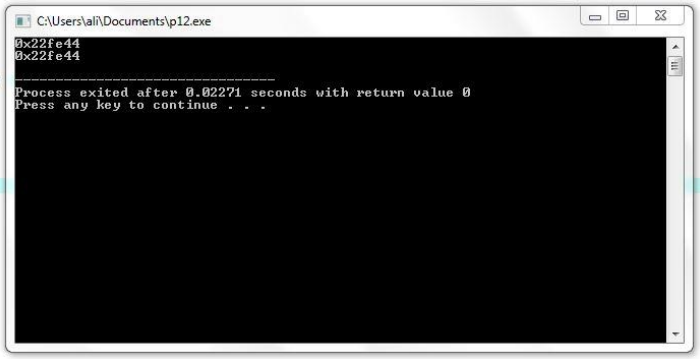



**Program 12:**

```
#include <iostream>
Using namespace std;
Const int MAX = 3;
Int main () {
  Int  var[MAX] = {10, 100, 200};
  Int  *ptr;
  // let us have address of the last element in pointer.
  Ptr = &var[MAX-1];
```

```
  For (int I = MAX; I > 0; i--) {
    Cout << “Address of var[“ << I << “] = “;
    Cout << ptr << endl;
    Cout << “Value of var[“ << I << “] = “;
    Cout << *ptr << endl;
    // point to the previous location
    Ptr--;
  }
  Return 0;
}
```

## Output:

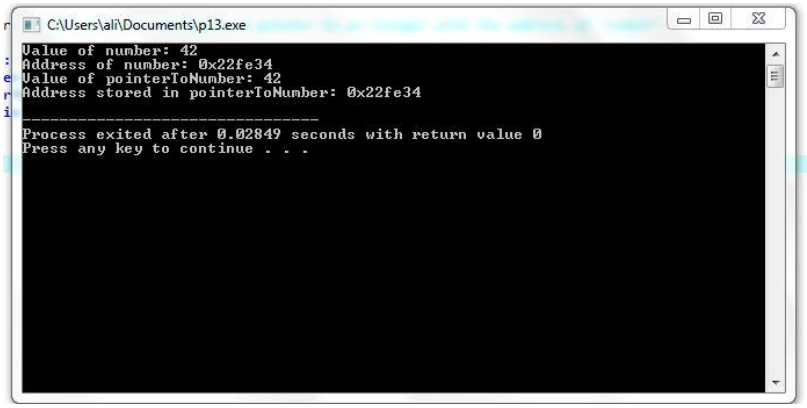



## Program 13:

```
#include <iostream>


int main() {
   int number = 42;
   int* pointerToNumber = &number; // Declare and initialize a pointer to an integer with the address of 'number'
```

```
    std::cout << "Value of number: " << number << std::endl;
    std::cout << "Address of number: " << &number << std::endl;
    std::cout << "Value of pointerToNumber: " << *pointerToNumber << std::endl;
    std::cout << "Address stored in pointerToNumber: " << pointerToNumber << std::endl;

    return 0;
}
```

**Output:**

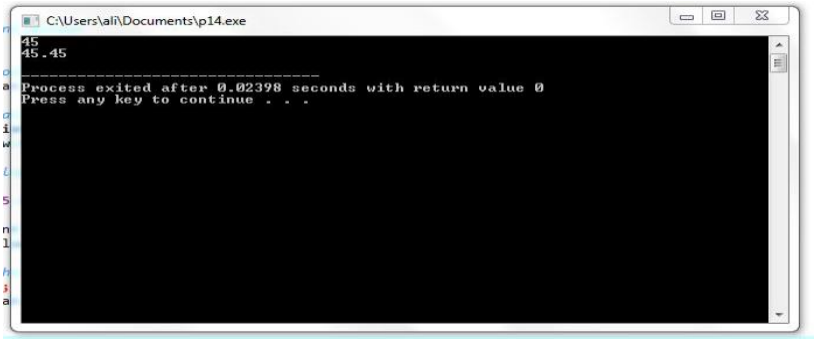



**Program 14:**

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

int main() {

  // declare an int pointer
  int* pointInt;
```

```
  // declare a float pointer
  float* pointFloat;

  // dynamically allocate memory
  pointInt = new int;
  pointFloat = new float;

  // assigning value to the memory
  *pointInt = 45;
  *pointFloat = 45.45f;

  cout << *pointInt << endl;
  cout << *pointFloat << endl;

  // deallocate the memory
  delete pointInt;
  delete pointFloat;

  return 0;
}
```

**Output:**




## Program 15:

```
// C++ Program to store GPA of n number of students and display it
// where n is the number of students entered by the user

#include <iostream>
using namespace std;

int main() {

 int num;
 cout << "Enter total number of students: ";
 cin >> num;
 float* ptr;

 // memory allocation of num number of floats
 ptr = new float[num];

 cout << "Enter GPA of students." << endl;
 for (int i = 0; i < num; ++i) {
  cout << "Student" << i + 1 << ": ";
```

```
  cin >> *(ptr + i);
 }
 cout << "\nDisplaying GPA of students." << endl;
 for (int i = 0; i < num; ++i) {
  cout << "Student" << i + 1 << ": " << *(ptr + i) << endl;
 }

 // ptr memory is released
 delete[] ptr;

 return 0;
}
```

**Output:**