void pointer\_to\_const\_int()

{

const int num=2;

// num++;//error

//pointer to const int

//The value in the pointer variable can be changed by

//assigning some other address to it. But the contents of the variable that it is

//pointing to cannot be modified through that pointer.

int num1 = 10;

const int \* ptr = &num1; // ptr is a pointer to a constant integer.

cout<<" ptr "<<ptr<<" value:"<<\*ptr<<endl;

int num2=90;

ptr = &num2; //Allowed

cout<<" ptr "<<ptr<<" value:"<<\*ptr<<endl;

ptr++; //Allowed

cout<<" ptr "<<ptr<<" value:"<<\*ptr<<endl;

//(\*ptr)++; //Not allowed:not allowed to change value

//\*ptr = 20; //Not allowed

}

//Constant pointer to an integer

void Constant\_pointer\_to\_integer()

{

int a=10;

int\* const ptr=&a;

cout<<"address :"<<ptr<<" value:"<<\*ptr<<endl;

cout<<"change value "<<endl;

\*ptr=900;

cout<<"address :"<<ptr<<" value:"<<\*ptr<<endl;

cout<<"change Address "<<endl;

int num=23;

//ptr=&num;//error :not allowed to change address

//ptr++;//errro:not allowed

}

Constant pointer to a constant integer

void Constant\_pointer\_constant\_integer()

{

//In this case, neither the address in the pointer be changed nor the contents at that

//address can be modified using that pointer

int num1=200;

const int\* const ptr=&num1;

cout<<" addresss "<<ptr<<" value:"<<\*ptr<<endl;

// ptr++;//error:not allowed

int num2=22;

//ptr=&num2;//not allowed

//\*ptr=100;//not allowed

}