```
//*************************************************************
//In the beginning on your answer sheet, please acknowledge that
// (i) no compilers or internet search are used in the test,
// (ii) only the (e)textbooks and your own notes are used, and
// (iii) you will not reveal this problem set to any other people.
//➔
//Violation is viewed as cheating.
//*************************************************************
//
//Record your answers in a WORD document and submit it under Canvas
//     in the end.
//Please write down in the WORD document
//     the output(s) that would be generated
//     by the main function
//     when it calls each of following functions one by one.

       //recursionf1_Demo();
       //recursionf2_DemoA();
       //recursionf2_DemoB();
       //exchange1ADemo();
       //exchange1BDemo();
       //exchange2ADemo();
       //exchange2BDemo();
       //pointerDemo();
       //vectorIteraterDemo();

//*****************************************************************************
// The program:
//*****************************************************************************
#include <iostream>         //inclusion of support for doing input & output
#include <vector>           //inclusion of support for doing input & output
using namespace std;        //declare access to standard stuff like cin, cout

//*************************************************************
// recursion
//*************************************************************
int f1( int n)
{      if (n <= 0)
              return 0;
       else
              return n*n + f1(n-1);
}

void recursionf1_Demo()
{
       cout << endl << endl
              << "*****************************************" << endl << endl
              << endl << endl ;
       //*************************************************************
       //On your answer sheet, describe what should be the output below.
       //*************************************************************
       cout << "recursion_f1_Demo(): " << endl;
       for (int i=0; i<5; i++)
              cout  << f1(i) << endl;

       cout << endl;
}
```

```
int f2( vector<int> & vec, int i, int j)
{
	//Base case
	if ( i == j)
		return vec[i];

	//Divide-and-conquer recursion below:
	int middle;
	middle = (i+j)/2;

	int x, y;
	x = f2(vec, i, middle);
	y = f2(vec, middle+1, j);

	if (x>y)
		return x;
	else
		return y;
}


void recursionf2_DemoA()
{
	cout << endl << endl
		 << "*****************************************" << endl << endl
		 << endl << endl ;

	vector<int> v;
	v.resize(5);

	//*****************************************************************
	//     Initialization:
	//*****************************************************************
	for (int i=0; i<5; i++)
		v[i] = i;

	//*****************************************************************
	//On your answer sheet, describe what should be the output below.
	//*****************************************************************
	cout << "recursion_f2_DemoA():Part I: " << endl;
	for (int i=0; i<5; i++)
		cout  << f2(v, i, i) << endl;

	cout << endl;

	//*****************************************************************
	//On your answer sheet, describe what should be the output below.
	//*****************************************************************
	cout << "recursion_f2_DemoA():Part II: " << endl;
	for (int i=0; i<4; i++)
		cout  << f2(v, i, i+1) << endl;

	cout << endl;
```

```
	//*****************************************************************
	//On your answer sheet, describe what should be the output below.
	//*****************************************************************
	cout << "recursion_f2_DemoA():Part III: " << endl;
	for (int i=0; i<3; i++)
		cout  << f2(v, i, i+2) << endl;

	cout << endl;


	//*****************************************************************
	//On your answer sheet, describe what should be the output below.
	//*****************************************************************
	cout << "recursion_f2_DemoA():Part IV: " << endl;
	for (int i=0; i<2; i++)
		cout  << f2(v, i, i+3) << endl;

	cout << endl;
}

void recursionf2_DemoB()
{
	cout << endl << endl
		 << "*****************************************" << endl << endl
		 << endl << endl ;

	vector<int> v;
	v.resize(5);

	//*****************************************************************
	//     Initialization:
	//*****************************************************************
	for (int i=0; i<5; i++)
		v[i] = 4-i;

	//*****************************************************************
	//On your answer sheet, describe what should be the output below.
	//*****************************************************************
	cout << "recursion_f2_DemoB: Part I: " << endl;
	for (int i=0; i<5; i++)
		cout  << f2(v, i, i) << endl;

	cout << endl;

	//*****************************************************************
	//On your answer sheet, describe what should be the output below.
	//*****************************************************************
	cout << "recursion_f2_DemoB: Part II: " << endl;
	for (int i=0; i<4; i++)
		cout  << f2(v, i, i+1) << endl;

	cout << endl;
```

```
	//*************************************************************
	//On your answer sheet, describe what should be the output below.
	//*************************************************************
	cout << "recursion_f2_DemoB: Part III: " << endl;
	for (int i=0; i<3; i++)
		cout  << f2(v, i, i+2) << endl;

	cout << endl;


	//*************************************************************
	//On your answer sheet, describe what should be the output below.
	//*************************************************************
	cout << "recursion_f2_DemoB: Part IV: " << endl;
	for (int i=0; i<2; i++)
		cout  << f2(v, i, i+3) << endl;

	cout << endl;
}



//*********************************************************
//Reference parameters, reference types, and pointers
//*********************************************************

void exchange1A(int & x, int  y)
{
	int   temp;
	temp = x;
	x = y;
	y = temp;
}

void exchange1ADemo()
{
	cout << endl << endl
		<< "*****************************************" << endl << endl
		<< endl << endl ;

	int p=100, q=0;
	exchange1A(p, q);


	//*************************************************************
	//On your answer sheet, describe what should be the output below.
	//*************************************************************
	cout << "exchange1ADemo(): " << endl;
	cout << "p = " << p << ", q = " << q << endl;

}
```

```
void exchange1B(int  x, int & y)
{
	int   temp;
	temp = x;
	x = y;
	y = temp;
}

void exchange1BDemo()
{
	cout << endl << endl
		<< "*****************************************" << endl << endl
		<< endl << endl ;

	int p=100, q=0;
	exchange1B(p, q);
	//*****************************************************************
	//On your answer sheet, describe what should be the output below.
	//*****************************************************************
	cout << "exchange1BDemo():" << endl;
	cout << "p = " << p << ", q = " << q << endl;

}


void exchange2A(int * x, int * y)
{
	int *  tempPtr;
	tempPtr = x;
	x = y;
	y = tempPtr;
}


void exchange2ADemo()
{
	cout << endl << endl
		<< "*****************************************" << endl << endl
		<< endl << endl ;

	int p=100, q=0;
	exchange2A(&p, &q);

	//*****************************************************************
	//On your answer sheet, describe what should be the output below.
	//*****************************************************************
	cout << "exchange2ADemo():" << endl;
	cout << "p = " << p << ", q = " << q << endl;

}
```

```
void exchange2B(int * x, int * y)
{
	int *  tempPtr;
	tempPtr = x;
	x = y;
	y = tempPtr;

	int   temp;
	temp = *x;
	*x = *y;
	*y = temp;
}


void exchange2BDemo()
{
	cout << endl << endl
		<< "*****************************************" << endl << endl
		<< endl << endl ;

	int p=100, q=0;
	exchange2B(&p, &q);
	//*****************************************************************
	//On your answer sheet, describe what should be the output below.
	//*****************************************************************
	cout << "exchange2BDemo():" << endl;
	cout << "p = " << p << ", q = " << q << endl;

}


//*************************************************************
// Pointers and memory allocation
//*************************************************************
void pointerDemo()
{
	cout << endl << endl
		<< "*****************************************" << endl << endl
		<< endl << endl ;

	int size =10;

	int * ptr;
	ptr = new int [size];

	//Storage allocated!
	//Store values into the vectos:
	for (int i=0; i< size; i++)
		ptr[i] = 0;

	int * itr;
	itr = ptr;

	*itr= 1;

	itr = itr+5;
	*itr = 1;
```

```
	itr = itr-3;
	*itr = 1;

	itr++;
	*itr = 1;

	itr[5] = 1;
	*(itr+6) = 1;

	//*************************************************************
	//On your answer sheet, describe what should be the output below.
	//*************************************************************
	cout << "pointerDemo(): " << endl;
	for (int i=0; i< size; i++)
		cout <<  ptr[i] << endl;


}



//*********************************************************
//vectors and the use of iteraters
//*********************************************************
void vectorIteraterDemo()
{	int size;
	cout << endl << endl
		<< "*******************************************" << endl << endl
		<< endl << endl ;


	vector<int>  v;
	v.resize(10);

	//Storage allocated!
	//Store values into the vectos:
	for (  vector<int>::iterator itr= v.begin();
			itr<v.end();
			itr++
	    )
	{
		*itr = 0;
	}

	v.insert( v.begin()+2, 1);
	v.insert( v.end(), 1);
	v.erase( v.begin(), v.begin()+2);

	//*************************************************************
	//On your answer sheet, describe what should be the output below.
	//*************************************************************
	cout << "iteraterDemo(): Part I: " << endl;
	for (  vector<int>::iterator itr= v.begin();
			itr<v.end();
			itr++
	     )
	{
```

```
		cout << *itr << endl;
	}


	vector<int>::iterator itr;
	itr= v.begin();

	*(itr+2) = 1;

	itr++;
	*itr = 1;

	itr = itr + 5;
	*itr = 1;

	itr = itr - 3;
	*itr = 1;

	itr++;
	*itr = 1;


	//*****************************************************************
	//On your answer sheet, describe what should be the output below.
	//*****************************************************************
	cout << endl << endl
		 << "iteraterDemo(): Part II: " << endl;
	for (  int i = 0;
		       i<v.size();
		       i++
		)
	{	cout << v[i]<< endl;
	}

}


int main()
{
	recursionf1_Demo();
	recursionf2_DemoA();
	recursionf2_DemoB();

	exchange1ADemo();
	exchange1BDemo();
	exchange2ADemo();
	exchange2BDemo();

	pointerDemo();
	vectorIteraterDemo();

	// Wait for the user input before ending the program
	char inputCharacter;
	cin >>  inputCharacter;

	//Finish the program and return the control to the operating system.
	return 0;
}
```