**PS2-Week 1- 2 Qsn 1**

#include <iostream>

#include <string>

using namespace std;

const int NUM = 5;

void pr(string str);

void prn();

typedef string prstr;

void populateArray(int arr[], int \*len);

void findIdx(int arr[], int aSz);

int main()

{

int len;

int arr[len];

populateArray(arr, &len);

findIdx(arr, len);

pr("\nPS2 Qsn1 Completed!");

prn();

}

void populateArray(int arr[], int \*len)

{

int theSize;

pr("\nEnter no of entries for array : ");

cin >> theSize;

\*len = theSize;

pr("\nEnter the Array values : ");

for (int i = 0; i < theSize; i++)

{

cin >> arr[i];

}

pr("\nArray values :");

for (int i = 0; i < theSize; i++)

{

pr(to\_string(arr[i]) + ",");

}

}

void findIdx(int arr[], int aSz)

{

int indx = 0, theNo;

int theSize = aSz;

pr("\nEnter the array value whose index is to be found : ");

cin >> theNo;

pr("\n Number to find index of : " + to\_string(theNo));

for (int i = 0; i < theSize; i++)

{

if (theNo == arr[i])

{

indx = i;

pr("\nThe Element : " + to\_string(arr[i]) + " exists at index : " + to\_string(i) + "\n");

return;

}

else if (i == (theSize - 1))

{

pr("\nNumber not in array\n");

};

}

}

void pr(string str)

{ cout << str; }

void prn()

{ cout << endl; }

**PS2-Week 1- 2 Qsn 2**

#include <iostream>

#include <string>

using namespace std;

const int NUM = 5;

void pr(string str);

void prn();

typedef string prstr;

void populateArray(int arr[], int \*len);

void findMaxMin(int arr[], int aSz, int \*max, int \*min);

void prArrayValues(int arr[], int len);

int main()

{

int len;

int arr[] = {23, 24, 3, 45, 10, 6, 3};

len = sizeof(arr) / sizeof(int);

prArrayValues(arr, len);

int min, max;

findMaxMin(arr, len, &max, &min);

pr("\nMaximum number in the array : " + to\_string(max));

pr("\nMinimum number in the array : " + to\_string(min));

pr("\nPS2 Qsn2 Completed!");

prn();

}

void populateArray(int arr[], int \*len)

{

int theSize;

pr("\nEnter no of entries for array : ");

cin >> theSize;

\*len = theSize;

pr("\nEnter the Array values : ");

for (int i = 0; i < theSize; i++)

{

cin >> arr[i];

}

pr("\nArray values : ");

for (int i = 0; i < theSize; i++)

{

pr(to\_string(arr[i]) + ",");

}

}

void prArrayValues(int arr[], int len)

{

int theSize = len;

pr("\nArray values : ");

for (int i = 0; i < theSize; i++)

{

pr(to\_string(arr[i]) + ",");

}

}

void findMaxMin(int arr[], int aSz, int \*max, int \*min)

{

int theSize = aSz;

int maxNo = arr[0], minNo = arr[0];

for (int i = 0; i < theSize; i++)

{

// pr("\nMaximum number in the array : " + to\_string(maxNo));

if (maxNo < arr[i])

{

maxNo = arr[i];

// pr("\n Cur Max No : " + to\_string(maxNo));

}

// pr("\nArr Val being tested : " + to\_string(arr[i]) + " against Min value : " + to\_string(minNo));

if (minNo > arr[i])

{

minNo = arr[i];

}

}

\*max = maxNo;

\*min = minNo;

}

void pr(string str)

{

cout << str;

}

void prn()

{

cout << endl;

}

**PS2-Week 1- 2 Qsn 3**

#include <iostream>

#include <string>

using namespace std;

const int NUM = 5;

void pr(string str);

void prn();

typedef string prstr;

void doLeftShift(int \*&arr, int len, int aSh);

void populateArray(int \*&arr, int \*len);

void prArrayValues(int \*arr, int len);

int main()

{

int \*arr, theSize, theOrigSh, theSh;

populateArray(arr, &theSize);

pr("\nEnter the no of left shifts to be done : ");

cin >> theOrigSh;

prArrayValues(arr, theSize);

theSh = theOrigSh % theSize;

doLeftShift(arr, theSize, theSh);

pr("\nUpdated Array after " + to\_string(theOrigSh) + " Left shifts");

prArrayValues(arr, theSize);

pr("\nPS2 Qsn3 Completed!");

prn();

}

void doLeftShift(int \*&arr, int len, int aSh)

{

int \*shArr;

int theShift = aSh;

int theSize = len;

shArr = new int[aSh];

for (int i = 0; i < theShift; i++)

{

shArr[i] = arr[i];

}

for (int i = 0; i < theSize - theShift; i++)

{

arr[i] = arr[i + theShift];

}

int lastPt = theSize - theShift;

for (int i = 0; i < theShift; i++, lastPt++)

{

arr[lastPt] = shArr[i];

}

}

void populateArray(int \*&arr, int \*len)

{

int theSize;

pr("\nEnter the no of entries for array : ");

cin >> theSize;

arr = new int[theSize];

\*len = theSize;

pr("\nEnter the Array values : ");

for (int i = 0; i < theSize; i++)

{

cin >> arr[i];

}

}

void prArrayValues(int \*arr, int len)

{

int theSize = len;

pr("\nArray values : ");

for (int i = 0; i < theSize; i++)

{

pr(to\_string(arr[i]) + ",");

}

}

void pr(string str)

{

cout << str;

}

void prn()

{

cout << endl;

}

**PS2-Week 1- 2 Qsn 4**

#include <iostream>

#include <string>

using namespace std;

const int NUM = 5;

void pr(string str);

void prn();

typedef string prstr;

int convertToEven(int \*aNo);

void processHalfArrayValues(int \*arr, int aSz);

void addBalVal(int \*&arr, int aSz, int aVal);

void populateArray(int \*&arr, int \*aSz);

void prArrayValues(int \*arr, int len);

int main()

{

int \*arr, theSize;

populateArray(arr, &theSize);

pr("\nArray values are : ");

prArrayValues(arr, theSize);

pr("\nThe two Half arrays are : ");

processHalfArrayValues(arr, theSize);

pr("\nArray values after balancing : ");

prArrayValues(arr, theSize);

pr("\nPS2 Qsn4 Completed!");

prn();

}

void populateArray(int \*&arr, int \*aSz)

{

int theSize;

pr("\nEnter a Even number as the no of entries for array : ");

cin >> theSize;

theSize = convertToEven(&theSize);

arr = new int[theSize];

\*aSz = theSize;

pr("\nEnter the Array values : ");

for (int i = 0; i < theSize; i++)

{

cin >> arr[i];

}

}

int convertToEven(int \*aNo)

{

if (0 == \*aNo % 2)

{

return \*aNo;

}

else

{

return \*aNo + 1;

}

}

void processHalfArrayValues(int \*arr, int aSz)

{

int theSize = aSz;

int fiSum = 0, seSum = 0;

pr("\nFirst Array values : ");

for (int i = 0; i < theSize / 2; i++)

{

pr(to\_string(arr[i]) + ",");

fiSum += arr[i];

}

pr("\nSecond Array values : ");

for (int i = theSize / 2; i < theSize; i++)

{

pr(to\_string(arr[i]) + ",");

seSum += arr[i];

}

pr("\nFirst Sum : " + to\_string(fiSum));

pr("\nSecond Sum : " + to\_string(seSum));

int diff = fiSum - seSum;

if (0 != diff)

{

pr("\nThe two Half arraya are not balanced!");

pr("\nThe smallest positive value to balance the arrays is : " + to\_string(abs(diff)));

addBalVal(arr, theSize, diff);

}

else

{

pr("\nThe two Half arraya are balanced!");

}

}

void addBalVal(int \*&arr, int aSz, int aVal)

{

int theSize = aSz;

int theBalVal = aVal;

if (theBalVal < 0)

{

arr[0] += -1 \* theBalVal;

}

else

{

arr[theSize / 2] += theBalVal;

}

}

void prArrayValues(int \*arr, int aSz)

{

int theSize = aSz;

pr("\nArray values : ");

for (int i = 0; i < theSize; i++)

{

pr(to\_string(arr[i]) + ",");

}

}

void pr(string str)

{

cout << str;

}

void prn()

{

cout << endl;

}

**PS2-Week 1- 2 Qsn 5**

#include <iostream>

#include <string>

using namespace std;

const int NUM = 5;

void pr(string str);

void prn();

typedef string prstr;

void populateArray(int \*&arr, int \*aSz);

void getTheInputNos(int \*&arr, string s, string del, int aSz);

int getTheSumOfArrValues(int \*arr, int aSz);

void prArrayValues(int \*arr, int len);

int main()

{

int \*arr, theSize;

populateArray(arr, &theSize);

int theSum = getTheSumOfArrValues(arr, theSize);

pr("\nThe Sum of all entries = " + to\_string(theSum));

prn();

pr("\nPS2 Qsn5 Completed!");

}

void populateArray(int \*&arr, int \*aSz)

{

int theSize;

string theStr;

pr("\nEnter the no of entries in the array : ");

cin >> theSize;

arr = new int[theSize];

\*aSz = theSize;

pr("\nEnter the Array values : ");

cin.ignore();

getline(cin, theStr);

string del = " ";

getTheInputNos(arr, theStr, del, theSize);

}

void getTheInputNos(int \*&arr, string s, string del, int aSz)

{

int theSize = aSz;

arr = new int[theSize];

int end = s.find(del); // Use find function to find 1st position of delimiter.

int indx = 0;

string str;

while (end != -1)

{ // Loop until no delimiter is left in the string.

str = s.substr(0, end);

arr[indx] = stoi(str);

indx++;

s.erase(s.begin(), s.begin() + end + 1);

end = s.find(del);

}

str = s.substr(0, end);

arr[indx] = stoi(str);

indx++;

}

int getTheSumOfArrValues(int \*arr, int aSz)

{

int theSize = aSz;

int theSum = 0;

for (int i = 0; i < theSize; i++)

{

theSum += arr[i];

}

return theSum;

}

void prArrayValues(int \*arr, int aSz)

{

int theSize = aSz;

pr("\nArray values : ");

for (int i = 0; i < theSize; i++)

{

pr(to\_string(arr[i]) + ",");

}

}

void pr(string str)

{

cout << str;

}

void prn()

{

cout << endl;

}

**PS2-Week 1- 2 Qsn 6**

#include <iostream>

#include <string>

using namespace std;

const int NUM = 5;

void pr(string str);

void prn();

typedef string prstr;

void getTheInputNos(int \*a1, int \*a2, int \*a3, int \*a4);

int getTheLargestOfFour(int a1, int a2, int a3, int a4);

int main()

{

int a1, a2, a3, a4;

getTheInputNos(&a1, &a2, &a3, &a4);

int theLargestNo = getTheLargestOfFour(a1, a2, a3, a4);

pr("\nThe largest number is : " + to\_string(theLargestNo));

prn();

pr("\nPS2 Qsn6 Completed!");

}

void getTheInputNos(int \*a1, int \*a2, int \*a3, int \*a4)

{

pr("\nEnter the entries one by one : ");

cin >> \*a1 >> \*a2 >> \*a3 >> \*a4;

}

int getTheLargestOfFour(int a1, int a2, int a3, int a4)

{

int largest;

if (a1 >= a2)

{ // a1 >= a2

if (a1 >= a3)

{ // a1 > a3

if (a1 >= a4)

{ // a1 > a4

largest = a1;

}

else // a4 > a1

{

largest = a4;

}

}

else // a3 > a1

{

if (a3 >= a4)

{ // a3 > a4

largest = a3;

}

else // a4 > a3

{

largest = a4;

}

}

}

else // a2> a1

{

if (a2 >= a3)

{ // a2 > a3

if (a2 >= a4)

{ // a2 > a4

largest = a2;

}

else // a4 > a2

{

largest = a4;

}

}

else // a3 > a2

{

if (a3 >= a4)

{ // a3 > a4

largest = a3;

}

else // a4 > a3

{

largest = a4;

}

}

}

return largest;

}

void pr(string str)

{

cout << str;

}

void prn()

{

cout << endl;

}