**PS3-Week 3- 4 Qsn 1**

#include <iostream>

#include <string>

using namespace std;

const int MAXLEN = 100;

void pr(string str);

void prn();

typedef string prstr;

void getInputName(char \*aName, int &aLen);

int getVowelCount(char aName[], int aLen);

int main()

{

int nameLen = MAXLEN;

char theName[nameLen]; // char theName[100] is C-string that will store

// max 99 char or less and last 100th or

// last + 1 as '\0' to indicate end of c-string;

// char \*p = theName;

getInputName(theName, nameLen);

int theVowelCnt = getVowelCount(theName, nameLen);

prstr str = "\nThe Vowel count : " + to\_string(theVowelCnt);

pr(str);

pr("\n\nPS3 Qsn1 Completed!");

}

void getInputName(char \*aName, int &aLen)

{

int theLen = aLen;

int i = 0;

pr("\nEnter your name : ");

cin.getline(aName, theLen);

do

{

i++;

} while (aName[i] != '\0');

aLen = i;

// pr("\nLength of name : " + to\_string(aLen));

}

int getVowelCount(char aName[], int aLen)

{

int theLen = aLen;

int theCnt = 0;

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

{

if (aName[i] == 'a' || aName[i] == 'e' || aName[i] == 'i' || aName[i] == 'o' || aName[i] == 'u')

{

theCnt++;

}

}

return theCnt;

}

void pr(string str)

{

cout << str;

}

void prn()

{

cout << endl;

}

**PS3-Week 3- 4 Qsn 2**

#include <iostream>

#include <string>

using namespace std;

const int MAXLEN = 100;

void pr(string str);

void prn();

typedef string prstr;

void getInputName(char \*aName, int &aLen);

void printNameReverse(char aName[], int aLen);

void makeACopy(char \*aName, int &aLen,char \*aNameCpy, int &aLenCpy);

int main()

{

int nameLen = MAXLEN;

char theName[nameLen]; // char theName[100] is C-string that will store

// max 99 char or less and last 100th or

// last + 1 as '\0' to indicate end of c-string;

// char \*p = theName;

getInputName(theName, nameLen);

prn();

printNameReverse(theName, nameLen);

int nameLenCpy = nameLen;

char theNameCpy[nameLenCpy];

makeACopy(theName, nameLen, theNameCpy, nameLenCpy);

prn();

pr("\n Your copied name : ");

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

{

cout << theNameCpy[i];

}

// int theVowelCnt = getVowelCount(theName, nameLen);

// prstr str = "\nThe Vowel count : " + to\_string(theVowelCnt);

// pr(str);

pr("\n\nPS3 Qsn2 Completed!");

}

void makeACopy(char \*aName, int &aLen,char \*aNameCpy, int &aLenCpy)

{

int theLen = aLen;

int i = 0;

//cin.getline(aName, theLen);

do

{

aNameCpy[i] = aName[i];

i++;

} while (aName[i] != '\0');

aLenCpy = i;

}

void getInputName(char \*aName, int &aLen)

{

int theLen = aLen;

int i = 0;

pr("\nEnter your name : ");

cin.getline(aName, theLen);

do

{

i++;

} while (aName[i] != '\0');

aLen = i;

pr("\nLength of your name : " + to\_string(aLen));

}

void printNameReverse(char aName[], int aLen)

{

int theLen = aLen;

pr("\nYour name in reverse : ");

for (int i = theLen-1; i >=0; i--)

{

cout << (aName[i]);

}

// return theCnt;

}

void pr(string str)

{

cout << str;

}

void prn()

{

cout << endl;

}

**PS3-Week 3- 4 Qsn 3**

#include <iostream>

#include <string>

using namespace std;

const int MAXLEN = 100;

void pr(string str);

void prn();

typedef string prstr;

void getInputName(char \*aName, int &aLen);

void convertToUpper(char \*aName, int &aLen);

int main()

{

int theLen = MAXLEN;

char theName[theLen];

getInputName(theName, theLen);

convertToUpper(theName, theLen);

pr("\nYour name is : ");

int i =0;

do

{

cout << theName[i];

i++;

}while (theName[i] != '\0');

prn();

pr("The number of ch is : " + to\_string(theLen));

prn();

pr("\nPS3 Qsn3 Completed!");

}

void getInputName(char \*aName, int &aLen)

{

int theLen = aLen;

pr("\nEnter your name : ");

cin.getline(aName, theLen);

int i =0;

do

{

i++;

}while(aName[i]!= '\0');

// pr("The number of ch is : " + to\_string(i));

aLen = i;

}

void convertToUpper(char \*aName, int &aLen)

{

int theLen = aLen;

int i =0;

char aCh;

do

{

aCh = aName[i];

if(aCh != ' ')

aName[i] = toupper(aCh);

i++;

}while(aName[i]!= '\0');

pr("The number of ch is : " + to\_string(i));

}

void pr(string str)

{

cout << str;

}

void prn()

{

cout << endl;

}

**PS3-Week 3- 4 Qsn 4**

#include <iostream>

#include <string>

using namespace std;

const int MAXLEN = 150;

const string WHITESPACE = " ";

void pr(string str);

void prn();

typedef string prstr;

void getInputString(char \*aStr, int &aLen);

void clearExtraWhiteSpaces(string &aStr, int aStPosn, int aEdPosn);

void findAndClearWhiteSpace(string &aStr);

void trimString(string &aStr);

int main()

{

int theLen = MAXLEN;

char theCStr[theLen];

string theStr;

getInputString(theCStr, theLen);

prn();

theStr = string(theCStr);

pr(theStr);

prn();

trimString(theStr);

// prn();

// pr(theStr);

findAndClearWhiteSpace(theStr);

pr("\nThe sentence after removel of blank spaces : \n\n");

pr(theStr);

prn();

pr("\nPS3 Qsn4 Completed!");

}

void findAndClearWhiteSpace(string &aStr)

{

int wsStPosn = 0;

int wsEdPosn = 0;

string findStr = WHITESPACE;

wsStPosn = aStr.find(findStr, wsStPosn);

do

{

wsEdPosn = aStr.find\_first\_not\_of(findStr, wsStPosn);

clearExtraWhiteSpaces(aStr, wsStPosn, wsEdPosn-1);

wsStPosn+=1;

wsStPosn = aStr.find(findStr, wsStPosn);

} while (wsStPosn != string::npos);

}

void clearExtraWhiteSpaces(string &aStr, int aStPosn, int aEdPosn)

{

int theStPosn = aStPosn;

int theEndPosn = aEdPosn;

int theReplaceLen = theEndPosn - theStPosn + 1;

aStr = aStr.replace(theStPosn, theReplaceLen, WHITESPACE);

}

void trimString(string &aStr)

{

int wsStPosn = 0;

int wsEdPosn = 0;

string findStr = WHITESPACE;

wsEdPosn = aStr.find\_first\_not\_of(findStr);

prn();

if(wsEdPosn != string::npos)

{

aStr = aStr.substr(wsEdPosn);

// pr(aStr);

}

wsEdPosn = aStr.find\_last\_not\_of(findStr);

if(wsEdPosn != string::npos)

{

aStr = aStr.substr(0, wsEdPosn + 1);

}

}

void getInputString(char \*aStr, int &aLen)

{

int theLen = aLen;

pr("\nEnter the sentence with some blank spaces(Max of 149 char) : \n");

cin.getline(aStr, theLen);

int i = 0;

for (i = 0; aStr[i] != '\0'; i++)

{

// cout << aStr[i];

}

aLen = i;

}

void pr(string str)

{

cout << str;

}

void prn()

{

cout << endl;

}

**PS3-Week 3-4 Qsn 5**

#include <iostream>

#include <string>

using namespace std;

const int MAXLEN = 150;

const string WHITESPACE = " ";

void pr(string str);

void prn();

typedef string prstr;

void getInputString(char \*aStr, int &aLen, int &aWdth);

void trimString(string &aStr);;

void segmentTheSentence(string &aStr, int aWdth);

int main()

{

int theLen = MAXLEN, theWidth = 0;

char theCStr[theLen];

string theStr;

getInputString(theCStr, theLen, theWidth);

prn();

theStr = string(theCStr);

trimString(theStr);

segmentTheSentence(theStr, theWidth);

prn();

pr("\nPS3 Qsn4 Completed!");

}

void getInputString(char \*aStr, int &aLen, int &aWdth)

{

int theLen = aLen;

int theWidth = aWdth;

pr("\nEnter a sentence (Max of 149 char) : ");

cin.getline(aStr, theLen);

pr("\nEnter the width of segments to break the sentence into : ");

cin >> theWidth;

int i = 0;

for (i = 0; aStr[i] != '\0'; i++)

{

// cout << aStr[i];

}

aLen = i;

aWdth = theWidth;

}

void trimString(string &aStr)

{

int wsStPosn = 0;

int wsEdPosn = 0;

string findStr = WHITESPACE;

wsEdPosn = aStr.find\_first\_not\_of(findStr);

prn();

if(wsEdPosn != string::npos)

{

aStr = aStr.substr(wsEdPosn);

// pr(aStr);

}

wsEdPosn = aStr.find\_last\_not\_of(findStr);

if(wsEdPosn != string::npos)

{

aStr = aStr.substr(0, wsEdPosn + 1);

}

}

void segmentTheSentence(string &aStr, int aWdth)

{

int theStPosn = 0;

int theWidth = aWdth;

string segStr = " ";

do

{

segStr = aStr.substr(theStPosn, theWidth);

pr(segStr);

prn();

theStPosn +=theWidth;

} while (theStPosn <= aStr.size());

}

void pr(string str)

{

cout << str;

}

void prn()

{

cout << endl;

}