[Ravi Patel] Instructor: Dr. Hindo

[CPSC 230]

Chapter 7 - lab assignment

(15 points)

Due date: End of this class

Note:

Save your document file as your lastname\_firstname

Submit your assignment in the inbox (chapter 7 assignment part 1).

Write your code and output for each program.

**Q1- What is the output**

int array[10]={0};

array[0]=10;

int i=1;

while (i< 10)

{ array[i] +=array[i-1] +2; //what is this statement doing?

cout <<array[i];

i++;

}

OUTPUT:

**121416182022242628**

//array[i] +=array[i-1] +2; //what is this statement doing?

This statement is assigning array[i] to array[i] plus array[i-1] plus 2. For example, since we set array[0] to 10, the first assignment is 10, then we add 2 to get 12. In the next iteration, Array[0] == 12, and we add 2 to it again to get array[1] == 14, and it keeps going until int i == 9

**Q2- Write a function to find the difference of two consecutive values in array ‘a’. Write the following functions (this is the formal way to write programs).**

**void introduction() ; // to display a text to tell what is the program about**

**bool keepRunning(); // to ask the user for more times to run the program**

**void runProgram(); // to input an array, call difference function and print the**

**result**

**void difference ( ) //diff is an array that saves the difference values**

#include <iostream>

//CPSC 230 RAVI PATEL DIFFERENCE USING ARRAY

using namespace std;

void introduction();

void newline();

void runProgram();

void display();

void difference();

void keepRunning();

//int a[10] = {0, 1, 10, 20, 30, 100, 50, 30, 80};

int diff[10];

double a[10];

int main() {

introduction();

newline();

runProgram();

newline();

display();

newline();

difference();

keepRunning();

newline();

}

void introduction(){

cout<<"This program will allow you to find the difference of two consecutive values in an array."<<endl;

}

void newline() {

int n;

for (n = 0; n < 1; n++)

printf("\n");

}

void runProgram(){

for (int j = 0; j < 10; j++){

cout<<"Enter a number for the array: ";

cin>>a[j];

}

}

void display(){

cout<<"a[10]"<<endl;

for (int i = 0; i<10; i++){

cout<<a[i]<<endl;

}

}

void difference(){

cout<<"diff[10]"<<endl;

for (int i = 0; i < 9; i++)

{

diff[i] = a[i + 1] - a[i];

cout<<diff[i]<<endl;

}

}

void keepRunning(){

char ch;

cout<<"Enter 'y' to continue the program: ";

cin>>ch;

while (ch == 'y' || ch == 'Y')

{

main();

}

exit(0);

}

**Sample Output:**

**This program will allow you to find the difference of two consecutive values in an array.**

**Enter a number for the array: 0**

**Enter a number for the array: 1**

**Enter a number for the array: 10**

**Enter a number for the array: 20**

**Enter a number for the array: 30**

**Enter a number for the array: 100**

**Enter a number for the array: 50**

**Enter a number for the array: 30**

**Enter a number for the array: 80**

**Enter a number for the array: 90**

**a[10]**

**0**

**1**

**10**

**20**

**30**

**100**

**50**

**30**

**80**

**90**

**diff[10]**

**1**

**9**

**10**

**10**

**70**

**-50**

**-20**

**50**

**10**

**Enter 'y' to continue the program: y**

**This program will allow you to find the difference of two consecutive values in an array.**

**Enter a number for the array: 0**

**Enter a number for the array: 2**

**Enter a number for the array: 20**

**Enter a number for the array: 30**

**Enter a number for the array: 40**

**Enter a number for the array: 50**

**Enter a number for the array: 70**

**Enter a number for the array: 90**

**Enter a number for the array: 110**

**Enter a number for the array: 20**

**a[10]**

**0**

**2**

**20**

**30**

**40**

**50**

**70**

**90**

**110**

**20**

**diff[10]**

**2**

**18**

**10**

**10**

**10**

**20**

**20**

**20**

**-90**

**Enter 'y' to continue the program: n**

**Example of the output**

|  |  |
| --- | --- |
| **a[10]**  **1**  **10**  **20**  **30**  **100**  **50**  **30**  **80** | **diff[10]**  **0**  **9**  **10**  **10**  **70**  **-50**  **-20**  **50** |

**Q3- Roll a six-sided die 60000 times. Then show the output frequency elements 1-6 in tabular format. Use the following**

**srand( time( 0 ) ); // seed random-number generator**

**face =1 + rand() % 6 ; // generate rand face from 1:6**

**Hint: use array B[face] to generate six items and save the number of “face” occurrence (i.e. B[1], B[2], B[3] …..) Then print the output as following:**

**Face Frequency**

**1 10030**

**2 10040**

**3 9990**

**4 9800**

**5 10130**

**6 10010**

//CPSC 230 RAVI PATEL ASSIGNMENT 7 Q3DICE ROLLER

#include <iostream>

using namespace std;

int main(int argc, char \*argv[]) {

float six = 0, five = 0, four = 0, three = 0, two = 0, one = 0;

int face;

int i = 60000;

srand (time(0)); //initialized all vars

do {

face = (rand()%6) + 1; //roll dice and set result to face

if (face == 6) //if face == 6

six++; //add 1 to six counter

if (face == 5)

five++;

if (face == 4)

four++;

if (face == 3)

three++;

if (face == 2)

two++;

if (face == 1)

one++;

--i; //go down iterations from 60000 until 0

}

while (i>0);

int B[6] = {one, two, three, four, five, six};

cout<<"Face"<<"\t"<<"Frequency"<<endl;

for (int k = 1; k<7; k++)

{

cout<<k<<"\t"<<B[k-1]<<endl;

}

}

**OUTPUT:**

**Face Frequency**

**1 9935**

**2 10016**

**3 10070**

**4 10170**

**5 9890**

**6 9919**

**Q2: Sort a list of 10 numbers using bubble sort (descending order). How many times does the comparison step occur? How many times does the swap part occur? Is this an optimized program?**

//CPSC 230 RAVI PATEL DESCENDING BUBBLE ARRAY SORTER

#include <iostream>

using namespace std;

int sort(int \*array, int size); //declared sort() - used to sort array in descending order

void display(int array[], int size); //declared display() - used to cout arrays

int sort(int \*array, int size) {

int step = 0;

for (int i = 0; i < size; i++) { //sorting here

for (int j = 1; j < (size - i); j++)

{

step += 1;

if (array[j - 1] < array[j]) // comparing here, < for descending order

{

step += 3;

int temp = array[j - 1]; //swapping here

array[j - 1] = array[j];

array[j] = temp;

}

}

}

return step;

}

void display(int array[], int size) { //simple display function used to cout arrays

cout << "(" << array[0];

for (int i = 1; i < size; i++)

{

cout << "," << array[i];

}

cout << ")";

}

int main() {

int a1[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; //declaring sample arrays and values

int a2[] = {10, 5, 3, 9, 2, 8, 4, 7, 1, 6};

int a3[] = {10, 9, 8, 7, 6, 5, 4, 3, 2, 1};

int steps = 0; //counting steps for compare and swap declared here

int size = 0; //size of array declared for ordering

size = sizeof(a1) / sizeof(int);

cout << "Sample array #1: ";

display(a1, size); //display first array

cout << endl;

steps = sort(a1, size); //sorting

cout << "Array #1 In Descending Order: ";

display(a1, size); //displaying sorted array

cout <<endl<< steps - 45 << " Swaps &" << " " << steps - (steps - 45) << " Comparisons || " << steps << " total steps completed!" << endl;

cout<<endl;

size = sizeof(a2) / sizeof(int);

cout << "Sample array #2: ";

display(a2, size); //display second array

cout << endl;

steps = sort(a2, size); //sorting here

cout << "Array #2 In Descending Order: ";

display(a2, size); //displaying sorted array

cout <<endl<< steps - 45 << " Swaps &" << " " << steps - (steps - 45) << " Comparisons || " << steps << " total steps completed!" << endl;

cout<<endl;

size = sizeof(a3) / sizeof(int);

cout << "Sample array #3: ";

display(a3, size); //displaying third array

cout << endl;

steps = sort(a3, size); //sorting array here

cout << "Array #3 In Descending Order: ";

display(a3, size); //displaying sorted array

cout <<endl<< steps - 45 << " Swaps &" << " " << steps - (steps - 45) << " Comparisons || " << steps << " total steps completed!" << endl;

return 0; //catch all just in case

}

**PROGRAM OUTPUT:**

**Sample array #1: (1,2,3,4,5,6,7,8,9,10)**

**Array #1 In Descending Order: (10,9,8,7,6,5,4,3,2,1)**

**135 Swaps & 45 Comparisons || 180 total steps completed!**

**Sample array #2: (10,5,3,9,2,8,4,7,1,6)**

**Array #2 In Descending Order: (10,9,8,7,6,5,4,3,2,1)**

**48 Swaps & 45 Comparisons || 93 total steps completed!**

**Sample array #3: (10,9,8,7,6,5,4,3,2,1)**

**Array #3 In Descending Order: (10,9,8,7,6,5,4,3,2,1)**

**0 Swaps & 45 Comparisons || 45 total steps completed!**

**//This is not an efficient or optimized program. There are many other ways to sort lists that do not require so much memory usage and can get the job done effectively.**

**Q3: Sort a list of 10 words using bubble sort**

//CPSC 230 RAVI PATEL WORD LIST SORTER

#include <iostream>

using namespace std;

const int word\_count = 10; //we will use a set of 10 words

void sorted(string wordArray[],int n); //declare sorting function

int main() {

string WordArray[word\_count]; //declare wordArray

string user\_input; //declare user\_input for user

int i; //int i for use in for loops

for(i = 0 ; i < word\_count ; i++)

{

cout<<"Enter a word: "; //ask user input

cin>>user\_input; //assign user input

WordArray[i] = user\_input; //assign user input to the word array

}

sorted(WordArray,word\_count); //call sort function

cout<< endl << "Words sorted in alphabetical order: "; //display sorted words

for(i = 0 ; i < word\_count ; i++) //this for loop will compare word with null character, adding space in list

{

if( WordArray[i].compare("\0") != 0 ) //compare with null character

cout << WordArray[i] << " "; //if null character, output space for formatting

}

return 0; //catch all

}

void sorted(string wordArray[],int n) { //sorter function to sort words in array alphabetically

int least, i;

string compared\_word;

for(i = 0; i < (n -1); i++) {

least = i;

compared\_word = wordArray[i];

for(int j = i+1; j < n; j++)

{

if (wordArray[j].compare(compared\_word)<0) //comparison made here

{

compared\_word = wordArray[j];

least = j;

}

}

wordArray[least] = wordArray[i];

wordArray[i] = compared\_word;

}

}

**SAMPLE OUTPUT:**

**Enter a word: apple**

**Enter a word: boat**

**Enter a word: aardvark**

**Enter a word: cat**

**Enter a word: dog**

**Enter a word: mouse**

**Enter a word: happy**

**Enter a word: sad**

**Enter a word: male**

**Enter a word: female**

**Words sorted in alphabetical order: aardvark apple boat cat dog female happy male mouse sad**