מבוא לתכנות מונחה עצמים

סטודנטית 1: דליה ויליאם

סטודנט 2: גיא רחמים

//Dalya William & Guy Rahamim

#include<iostream>

#define END std::endl

#define END2 std::cout<<std::endl<<std::endl;

int countCharacterInstances(char\* str1, char\* str2);

int similarityIndex(char\* str1, char\* str2);

int encryptWord(char\* str1, char\* str2);

int\* firstEvenNumber(int\* array, int size);

int sortTheArray(int\* array, int size);

void printArray(int\* array, int size);

int main()

{

//Question 1---------------------------

char q1Str1[] = "aabbcdef";

char q1Str2[] = "abc";

std::cout << "Question 1-----------------------" << END;

std::cout << "values before function: " << q1Str1 << ",\t" << q1Str2 << END;

std::cout << "number of reoccuring characters: " << countCharacterInstances(q1Str1, q1Str2) << END;

END2

//Question 2---------------------------

std::cout << "Question 2-----------------------" << END;

char q2Str1[] = "keyboard";

char q2Str2[] = "blackboard";

std::cout << "values before function: " << q2Str1 << ",\t" << q2Str2 << END;

int similarityIndex=countCharacterInstances(q2Str1, q2Str2);

std::cout << "index of similarity: " <<similarityIndex<< END;

END2

//Question 3 --------------------------

std::cout << "Question 3----------------------- " <<END;

char q3Str1[] = "My hat has 3 corners, 3 corners my hat has\nif the hat hadn't had 3 corners, it would not be my hat";

char q3Str2[] = "hat";

int counter = 0;

std::cout << "values before function: " << END << q3Str1 << "\n" << q3Str2 << END;

counter =encryptWord(q3Str1, q3Str2);

std::cout << std::endl << "values after function: " << END << q3Str1 << END;

std::cout << std::endl << "number of encrypted Words: " << counter << END;

END2

//Question 4 --------------------------

std::cout << "Question 4----------------------- " << END;

int q4Array[] = { 3,4,5,7,2,5,3 };

int q4Size = 6;

std::cout << "values before function: " << END;

printArray(q4Array,q4Size);

std::cout << END << END << "first even numbers index: " << \*firstEvenNumber(q4Array, q4Size) << END;

END2

//Question 5---------------------------

std::cout << "Question 5----------------------- " << END;

int q5Array[] = {9,8,5,8,3,2,7,7,5,8,3};

int q5Size = 11;

std::cout << "array before function: " << END;

printArray(q5Array, q5Size);

std::cout << END;

int count= sortTheArray(q5Array, q5Size);

std::cout << "array after function: " << END;

printArray(q5Array, q5Size);

std::cout << END << END <<"number of distinct number in the array: " << count;

END2

return 1;

}

//Question 1 function

int countCharacterInstances(char\* str1, char\* str2)

{

int counter = 0;

int str2counter = 0;

while (\*str1 != NULL)

{

while (\*str2 != NULL)

{

if (\*str1 == \*str2)

{

counter++;

}

str2counter++;

str2++;

}

str2 -= str2counter;

str1++;

}

return counter;

}

//Question 2 function

int similarityIndex(char\* str1, char\* str2)

{

int result = -1;

int str1Length = strlen(str1);

int str2Length = strlen(str2);

for (int i = 0; i < str1Length; i++)

{

for (int j = 0; j < str2Length; j++)

{

if (\*str1 == \*str2)

{

if (strcmp(str1, str2) == 0)

return i;

}

str2++;

}

str1++;

str2 -= str2Length;

}

return result;

}

//Question 3 function

int encryptWord(char\* str1, char\* str2)

{

//initialize variables

char star = '\*';

int counter = 0;

int str2Length = strlen(str2);

//while str2 is a substring of str1

while (strstr(str1, str2) != NULL)

{

counter++;

//next index holds the distance between index 0 and the first

//occurence of str2 in str1.

int nextIndex = (int)strstr(str1, str2) - (int)str1;

//for loop run along each occurence of str2 in str1.

for (int i = nextIndex; i < nextIndex + str2Length; i++)

{ \*(str1 + i) = star; }

}

return counter;

}

//Question 4 function

int\* firstEvenNumber(int\* array, int size)

{

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

{

if (\*(array + i) % 2 == 0)

return (array+i);

}

return NULL;

}

//Question 5 function

int sortTheArray(int\* array, int size)

{

int tempSize = size;

int counter = 0;

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

{

for (int j = 0; j <size-1; j++)

{

if (\*(array + j) > \*(array + j + 1))

{

int temp = \*(array + j);

\*(array + j) = temp;

\*(array + j) = \*(array + j + 1);

\*(array + j + 1) = temp;

}

}

}

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

{

if (\*(array+i)!=\*(array+i+1))

counter++;

}

return counter;

}

//printing an array

void printArray(int\* array, int size)

{

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

{

std::cout << \*(array + i) << "\t";

}

}

Output:

