

Лабораторна робота №9. Рядки

Міністерство освіти і науки України
Національний технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації
і управління

Звіт

з лабораторної роботи № 9 з дисципліни
«Основи програмування»

«Рядки»

Варіант __4__

Виконав студент _____ Берлінський Ярослав Владленович _____
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2020

Назва роботи: робота з рядками.

Варіант: 4

Умова задачі:

4. Ввести два рядки, вилучити з першого рядка всі слова, які зустрічаються у другому рядку.

Постановка задачі. Користувачем записуються два рядки, програмою генеруються 2 масиви рядків(у даному випадку - зі словами), вилучених з обох рядків. Порівнюються слова. Зі слів, що належать першому рядку та не співпадають з елементами другого рядку, формується новий масив рядкового типу та виводиться користувачеві.

Розв'язок:

1. C++:

Main

```
LAB5 > LAB5 > main.cpp > main()
1 //
2 // main.cpp
3 // LAB8
4 //
5 // Created by Берлинский Ярослав Владленович on 11.11.2020.
6 #include <iostream>
7 #include <string>
8 #include "functional_part.cpp"
9 using namespace std;
10
11 int main(){
12     string words_from1[N];
13     string words_from2[N];
14     string final_str[N];
15     char str1[N];
16     char str2[N];
17     cout<<"Input the first string: ";
18     gets(str1);
19     cout<<"Input the second string: ";
20     gets(str2);
21     int i=0, j=0, index=0;
22     i = creating_words(str1, words_from1);
23     j = creating_words(str2, words_from2);
24     index = comparing_words(words_from1, words_from2, &i, &j, final_str);
25     final_output(final_str, &index);
26 }
27
28 //1: zero one two three four five six seven eight nine ten ten eleven kni
29 //2: zero one two three four five six seven eight nine ten ten eleven kni
```

Functional part

```
functional_part.cpp > comparing_words(words_from1, words_from2, pt1, pt2, arr3)
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 int N=1000;
5 int creating_words(char *str, string words_from1[]){
6     char *pch = strtok(str, " ,.-");
7     int i=0;
8     while (pch!=NULL){
9         words_from1[i]=pch;
10        i++;
11        pch = strtok(NULL, " ,.-");
12    }
13    //for (int j=0; j<i; j++){
14        // cout<<"["<<j<<" ] "<<words_from1[j]<<endl;
15    return i;
16 }
17 void final_output(string array[], int *index){
18     cout<<"\nNew string is here: ";
19     for (int i=0; i<*index; i++){
20         cout<<array[i]<<" ";
21     }
22     cout<<endl;
23 }
24 int comparing_words(string words_from1[], string words_from2[], int *pt1, int *pt2, string arr3[]){
25     int f=0, index_new=0;
26     for(int i=0; i<*pt1; i++){
27         for(int j=0; j<*pt2; j++){
28             if(words_from1[i]==words_from2[j]){
29                 f=1;
30             }
31         }
32     }
33     return f;
34 }
```

```
functional_part.cpp > comparing_words(words_from1, words_from2, pt1, pt2, arr3)

17 void final_output(string array[], int *index){
18     cout<<"\nNew string is here: ";
19     for (int i=0; i<*index; i++)
20         cout<<array[i]<<" ";
21     cout<<endl;
22 }
23 int comparing_words(string words_from1[], string words_from2[], int *pt1, int *pt2, string arr3[]){
24     int f=0, index_new=0;
25     for(int i=0; i<*pt1; i++){
26         for(int j=0; j<*pt2; j++){
27             if(words_from1[i]==words_from2[j]){
28                 f=1;
29             }
30         }
31         if (f==0){
32             arr3[index_new]=words_from1[i];
33             index_new++;
34         }
35         f=0;
36     }
37     N=index_new;
38     return index_new;
39 }
40
```

2. Python:

Main

```
main.py x functions.py x
from functions import creating_strings, comparing_words, final_output
row1=input("Input string1: ")
row2=input("Input string2: ")
#тут лежат сепараторы в списке
sep = [' ', ',', '.', ':', '-', '(', ')', '[', ']', '{', '}']
#####
string1, string2, fn_string = [], [], []
string1 = creating_strings(row1, string1, sep)
string2 = creating_strings(row2, string2, sep)
string3 = comparing_words(string1, string2)
final_output(string3)
```

Functional part

```
main.py x functions.py x
1 def creating_strings(list1, list2, sep):
2     word=""
3     for i in list1:
4         flag = 0
5         for j in sep:
6             if i==j:
7                 flag=1
8         if flag==0:
9             word+=i
10        else:
11            list2.append(word)
12            word=""
13        list2.append(word)
14    return list2
15 def comparing_words(list1, list2):
16     f=0; index_new=0; list3=[]
17     for word1 in list1:
18         for word2 in list2:
19             if word1==word2:
20                 f=1
21             if f==0:
22                 list3.append(word1)
23                 index_new+=1
24             f=0
25    return list3
26
27 def final_output(list):
28     for word in list:
29         print(word, end=" ")
```

Тестування вхідних даних(C++):

1)

```
Input the first string: warning: this program uses gets(), which is unsafe.  
zero one two three four five six seven eight nine ten ten eleven kni  
Input the second string: zero two six six six-five 098 kni  
  
New string is here: one three four seven eight nine ten ten eleven  
Program ended with exit code: 0
```

2)

```
Input the first string: warning: this program uses gets(), which is unsafe.  
живи люби kni  
Input the second string: люби живи 1234 ащмушим уоуршуцпуопршгунгу  
  
New string is here: kni  
Program ended with exit code: 0
```

Тестування вхідних даних(Python):

1)

```
Input string#1: zero one zero two zero three four five seven six eight seven nine ten eleven  
Input string#2: eleven zero six seven  
one two three four five eight nine ten
```

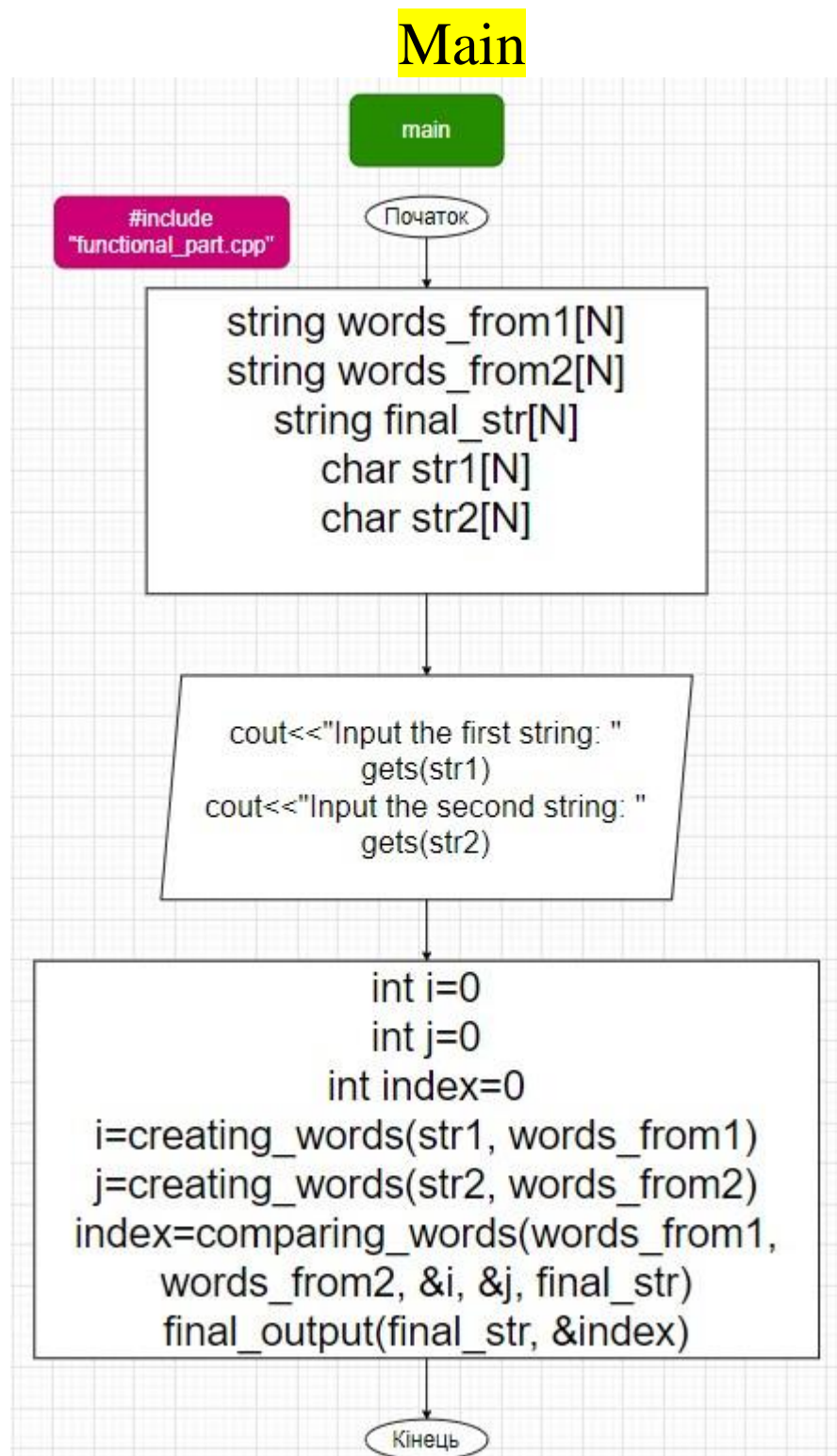
2)

```
C:\Users\Ярослав\AppData\Local\Programs\F  
Input string#1: 1 2 3 4 kni лдин два три  
Input string#2: два три  
1 2 3 4 kni лдин
```

Нескладно перевірити вірність результатів. Інші дані можна перевірити самотійно, перейшовши безпосередньо до коду програми:

[GitHub](#)

Блок-схема роботи алгоритму:

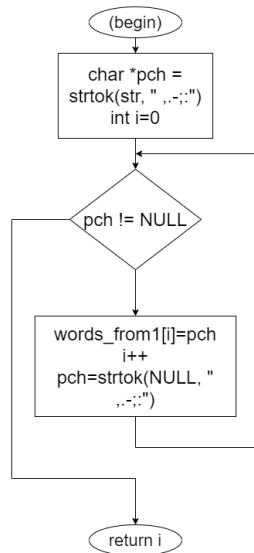


Functional_part

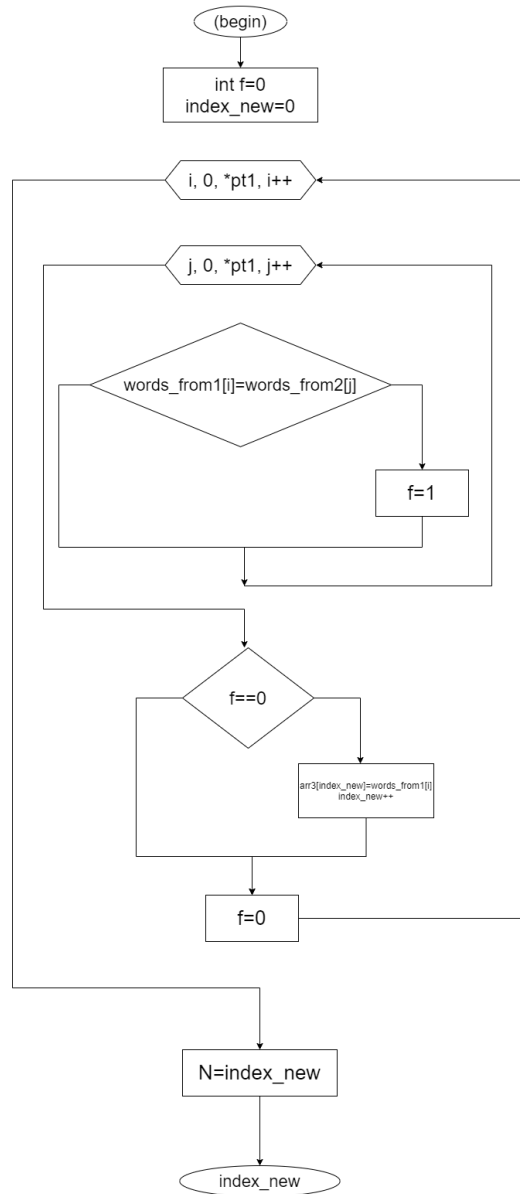
functional_part

N=1000

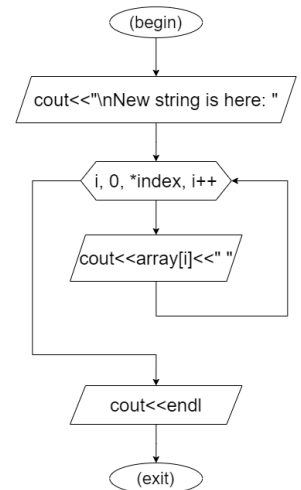
creating_words(char *str,
string words_from1[])



comparing_words(string words_from1[], string
words_from2[], int *pt1, int *pt2, string arr3[])



final_output(string array[], int *index)



functional_part

N=1000

creating_words(char *str,
string words_from1[])

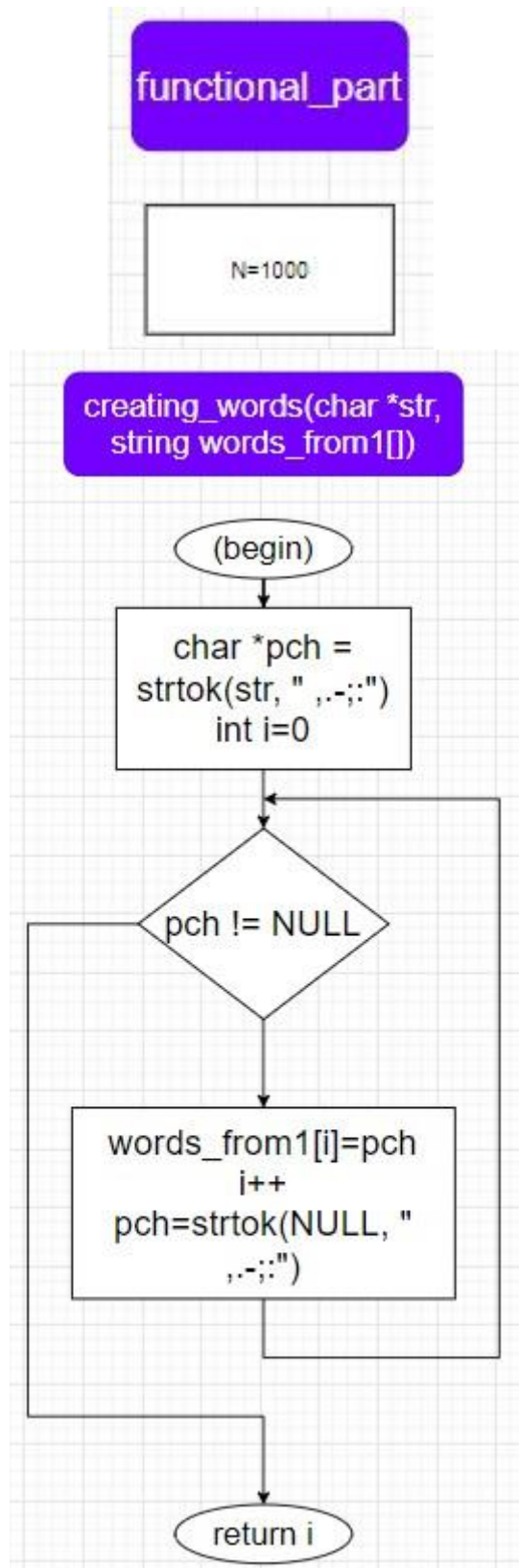
(begin)

char *pch =
strtok(str, " ,.-;:")
int i=0

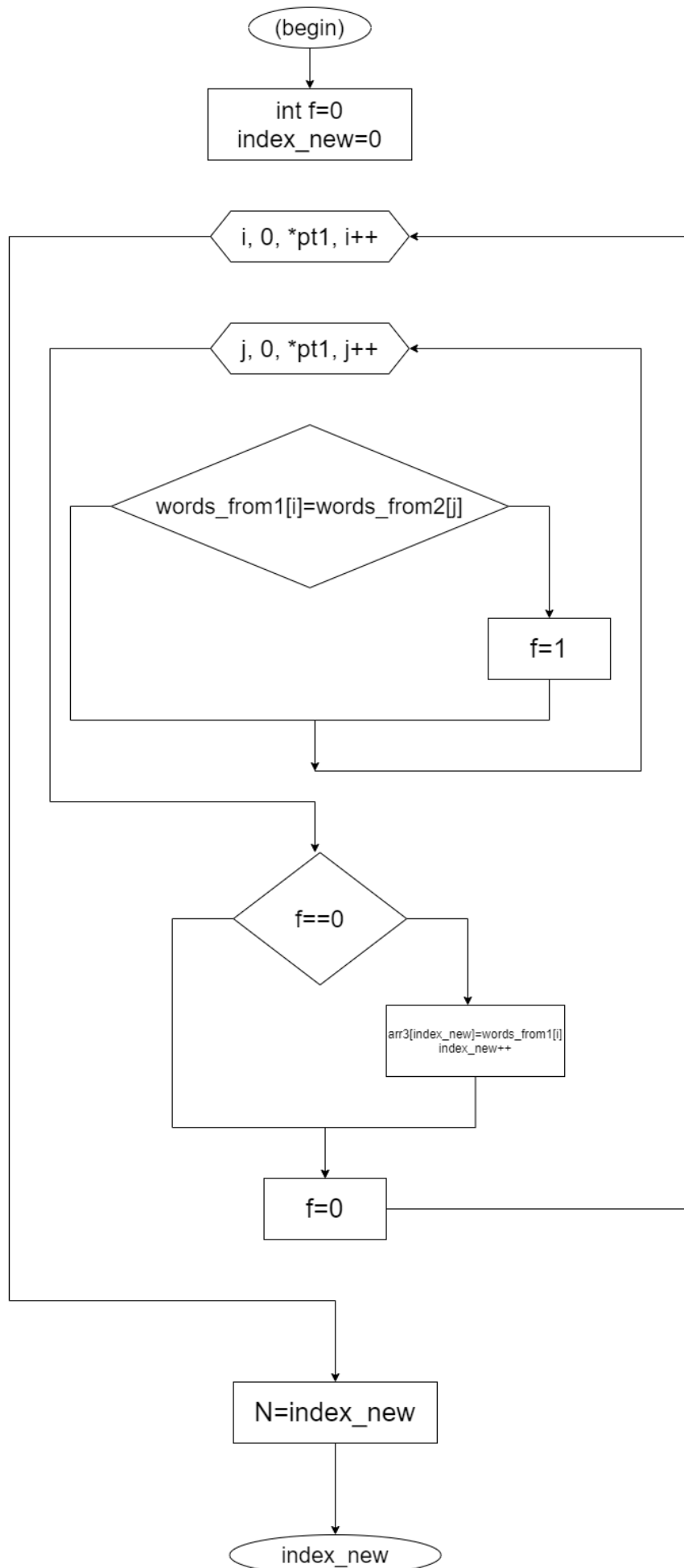
pch != NULL

words_from1[i]=pch
i++
pch=strtok(NULL, "
 ,.-;:")

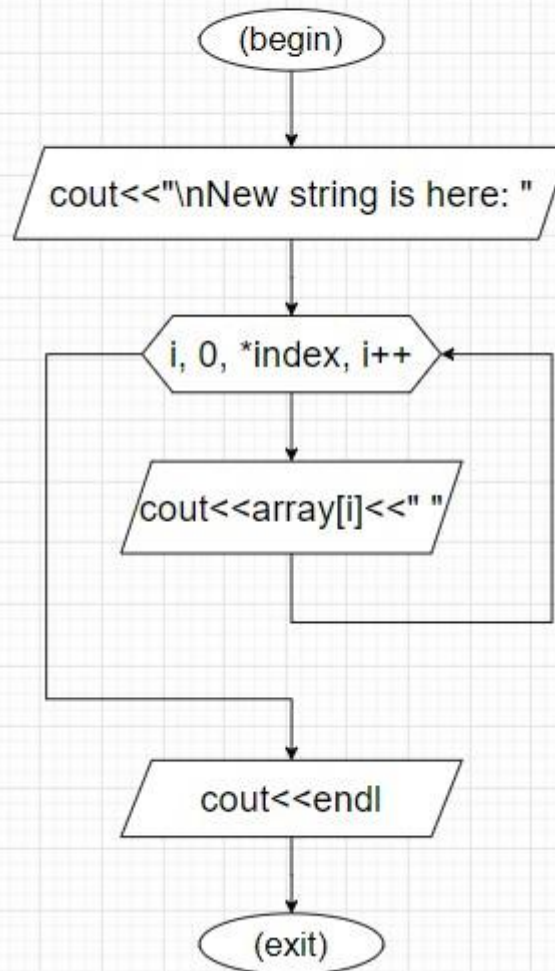
return i



comparing_words(string words_from1[], string words_from2[], int *pt1, int *pt2, string arr3[])



final_output(string array[], int *index)



Висновок: отже, програма була реалізована одним з оптимальних варіантів через роботу трьох масивів через використання арифметичних та ітераційних циклів. Дані передавалися покажчиками(C++). Також було реалізовано алгоритм та інтерпретованій мові(Python).