

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

Дисципліна: “Програмування та алгоритмічні мови”

Лабораторна робота № 1

Тема робіт: «Динамічні структури даних»

Варіант №18

Виконав: студент
1 курсу групи КА-95
Петренко Д.М.

Прийняв: Гуськова В.Г.

Мета роботи. Навчитися створювати функції і правильно користуватися динамічними структурами даних .

Умова задачі.

11 - 18

Оголосити структуру «Алфавіт», «букви» – як слово, складене з усіх букв даного алфавіту у лексикографічному порядку, аналогічно побудоване поле «знаки» – із усіх припустимих знаків, і два числових поля з інформацією про кількість букв і спеціальних символів даного алфавіту.

Оголосити вказівник на динамічний масив змінних такого структурного типу.

Передбачити такі функції:

1. створення масиву потрібного розміру з виділенням пам'яті і заповненням його вмісту з клавіатури;
2. виведення вмісту створеного масиву на екран;
3. підрахунок кількості елементів, алфавіт яких складений із заданої кількості символів (передати як параметр);
4. впорядкування елементів за кількістю букв;

PDF создан испытательной версией pdfFactory Pro www.pdffactory.com

5. додавання нового елемента після **k**-го (**k** передати як параметр)
6. поміняти місцями **k**-й і **j**-й елементи (**k** і **j** передати як параметр)

Створити двійковий файл, **k**-й запис якого міститиме інформацію про **k**-й елемент масиву. Передбачити функцію для заповнення цього файлу із масиву.

Код програми мовою C++:

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <fstream>
using namespace std;

struct Alphabet // оголашаем структуру
{
    char letters[40];
    char symbols[40];
```

```

    int lc;
    int sc;
}*p;

void base(int N) // создаём структуру
{
    p = new Alphabet[N];
    for (int i = 0; i < N; i++)
    {
        cout << "Enter letters of the alphabet\n"; // буквы
        cin >> p[i].letters;
        cout << "Enter special symbols of the alphabet\n"; //символы
        cin >> p[i].symbols;
        p[i].lc = strlen(p[i].letters); // количество букв и количество символов
        p[i].sc = strlen(p[i].symbols);
    }
}

void print(int N) //вывод
{
    for (int i = 0; i < N; i++)
    {
        cout <<i + 1 << ". Alphabet contains " << p[i].lc << " letters: " <<
p[i].letters << " and " << p[i].sc << " special symbols: " << p[i].symbols
<< "\n";

    }
}

int comb(int k, int N) // функция подсчёта элементов, что имеют определённое
количество букв и символов
{
    int m=0;
    for (int i = 0; i < N; i++)
    {
        if ((p[i].lc + p[i].sc) == k)
        {
            m++;
            cout << "i + 1 " << ".Alphabet contains " << p[i].lc << " letters: "
<< p[i].letters << " and " << p[i].sc << " special symbols : " <<
p[i].symbols << "\n";
        }
    }
    return m;
}

```

```

void how(int N) // функция для подсчёта элементов. Здесь ты только вводишь
данные
{
    int k,m;
    cout << "Enter how much symbols you require\n";
    cin >> k;
    m = comb(k, N);
    cout << "Total " << m << "\n";
}

void swap(int a, int b) //Функция меняет местами 2 элемента
{
    char temp1[40];
    char temp2[40];
    int t1, t2;
    strcpy(temp1, p[a].letters);
    strcpy(temp2, p[a].symbols);
    t1 = p[a].lc;
    t2 = p[a].sc;
    strcpy(p[a].letters, p[b].letters);
    strcpy(p[a].symbols, p[b].symbols);
    p[a].lc = p[b].lc;
    p[a].sc = p[b].sc;
    strcpy(p[b].letters, temp1);
    strcpy(p[b].symbols, temp2);
    p[b].lc = t1;
    p[b].sc = t2;
}

void sort(int N) //функция сортирует
{
    for (int j = 0; j<N; j++){
        for (int i = 1; i < N; i++)
        {
            if (p[i].lc < p[i-1].lc)
            {
                swap(i - 1, i); // переставляет элементы с помощью функции
перестановки
            }
        }
    }
    cout << "Base is sorted\n";
}

void addel(int N, int k) //функция которая добавляет элемент после k-го
{
    p=(Alphabet*)realloc(p,sizeof(Alphabet)*(N+1));

```

```

for (int i = N; i > k-1; i--) //Сдвиг всех элементов вправо
{
    strcpy(p[i].letters, p[i-1].letters);
    strcpy(p[i].symbols, p[i-1].symbols);
    p[i].lc = p[i-1].lc;
    p[i].sc = p[i-1].sc;
}
cout << "Enter letters of the alphabet\n";
cin >> p[k].letters;
cout << "Enter special sybols of the alphabet\n";
cin >> p[k].symbols;
p[k].lc = strlen(p[k].letters);
p[k].sc = strlen(p[k].symbols);
}

void changel(int N) //функция для для ввода элементов для замены
{
    int k, j;
    system("cls");
    cout << "Enter elements to swap\n";
    cin >> k;
    cin >> j;
    swap(k - 1, j - 1);
}

void adel(int N) // функция для ввода элемента, после которого добавляем
элемент
{
    int k;
    cout << "Enter number of element after which you want to add\n";
    cin >> k;
    addel(N, k);
}

void write(int n) // создание файла
{
    FILE *fp;
    if(!(fp=fopen("file.txt", "wb")))
    {
        printf("Cannot open file\n");
        return;
    }

    for(int i = 0; i < n; i++)
    {
        fwrite(&p[i], sizeof(Alphabet), 1, fp);
    }
}

```

```

    fclose(fp);
}
int main()
{
    int N,f;
    cout<<"Enter dimension of database.\n";
    cin>>N;

    base(N);
    do
    {
        print(N);
        cout << "    Menu    \n";
        cout << "1.Count alphabets with needed characters\n";
        cout << "2.Sort by letters number\n";
        cout << "3.Add new element after certain\n";
        cout << "4.Change place of two elements\n";
        cout << "5.Save data to the file\n";
        cout << "6.Exit\n";
        cin >> f;
        switch (f)
        {
            case 1: {how(N); break;}
            case 2: { sort(N); break;}
            case 3: { adel(N); N++; break;}
            case 4: { changel(N); break;}
            case 5: { write(N); break;}
            case 6: break;
        }
    } while (f != 6);
}

```

Контрольні приклади:

main.cpp

  saved

```

1  #include <iostream>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <stdio.h>
5  #include <fstream>
6  using namespace std;
7
8
9  struct Alphabet // оглашаем структуру
10 {
11     char letters[40];
12     char symbols[40];
13     int lc;
14     int sc;
15 }*p;
16
17 void base(int N) // создаём структуру
18 {
19     p = new Alphabet[N];
20     for (int i = 0; i < N; i++)
21     {
22         cout << "Enter letters of the alphabet\n"; // буквы
23         cin >> p[i].letters;
24         cout << "Enter special symbols of the alphabet\n"; //символы
25         cin >> p[i].symbols;
26         p[i].lc = strlen(p[i].letters); // количество букв и количество символов
27         p[i].sc = strlen(p[i].symbols);
28     }
29 }
30
31 void print(int N) //вывод
32 {
33     for (int i = 0; i < N; i++)
34     {
35         cout << i + 1 << ". Alphabet contains " << p[i].lc << " letters: " << p[i].letters << " and " << p[i].sc
36         << " special symbols: " << p[i].symbols << "\n";
37     }
38 }
39
40 int comb(int k, int N) // функция подсчёта элементов, что имеют определённое количество букв и символов
41 {
42     int m=0;
43     for (int i = 0; i < N; i++)
44     {
45         if ((p[i].lc + p[i].sc) == k)
46         {
47             m++;
48             cout << "i + 1 " << ".Alphabet contains " << p[i].lc << " letters: " << p[i].letters << " and " << p
49             [i].sc << " special symbols : " << p[i].symbols << "\n";
50         }
51     }
52     return m;
53 }
54
55
56 void how(int N) // функция для подсчёта элементов. Здесь ты только вводишь данные
57 {
58     int k,m;
59     ..

```

```

59     cout << "Enter how much symbols you require\n";
60     cin >> k;
61     m = comb(k, N);
62     cout << "Total " << m << "\n";
63 }
64
65 void swap(int a, int b) //Функция меняет местами 2 элемента
66 {
67     char temp1[40];
68     char temp2[40];
69     int t1, t2;
70     strcpy(temp1, p[a].letters);
71     strcpy(temp2, p[a].symbols);
72     t1 = p[a].lc;
73     t2 = p[a].sc;
74     strcpy(p[a].letters, p[b].letters);
75     strcpy(p[a].symbols, p[b].symbols);
76     p[a].lc = p[b].lc;
77     p[a].sc = p[b].sc;
78     strcpy(p[b].letters, temp1);
79     strcpy(p[b].symbols, temp2);
80     p[b].lc = t1;
81     p[b].sc = t2;
82 }
83
84 void sort(int N) //функция сортирует
85 {
86     for (int j = 0; j<N; j++){
87         for (int i = 1; i < N; i++)
88         {
89             // ...

```



```

89     if (p[i].lc < p[i-1].lc)
90     {
91         swap(i - 1, i); // переставляет элементы с помощью функции перестановки
92     }
93 }
94 }
95 cout << "Base is sorted\n";
96 }
97
98 void addel(int N, int k) //функция которая добавляет элемент после k-го
99 {
100     p=(Alphabet*)realloc(p,sizeof(Alphabet)*(N+1));
101     for (int i = N; i > k-1; i--) //Сдвиг всех элементов вправо
102     {
103         strcpy(p[i].letters, p[i-1].letters);
104         strcpy(p[i].symbols, p[i-1].symbols);
105         p[i].lc = p[i-1].lc;
106         p[i].sc = p[i-1].sc;
107     }
108     cout << "Enter letters of the alphabet\n";
109     cin >> p[k].letters;
110     cout << "Enter special sybols of the alphabet\n";
111     cin >> p[k].symbols;
112     p[k].lc = strlen(p[k].letters);
113     p[k].sc = strlen(p[k].symbols);
114 }
115
116 void changel(int N) //функция для для ввода элементов для замены
117 {

```

```

118     int k, j;
119     system("cls");
120     cout << "Enter elements to swap\n";
121     cin >> k;
122     cin >> j;
123     swap(k - 1, j - 1);
124 }
125
126 void adel(int N) // функция для ввода элемента, после которого добавляем элемент
127 {
128     int k;
129     cout << "Enter number of element after which you want to add\n";
130     cin >> k;
131     addel(N, k);
132 }
133
134 void write(int n) // создание файла
135 {
136     FILE *fp;
137     if(!(fp=fopen("file.txt", "wb")))
138     {
139         printf("Cannot open file\n");
140         return;
141     }
142
143     for(int i = 0; i < n; i++)
144     {
145         fwrite(&p[i], sizeof(Alphabet), 1, fp);
146     }
147     fclose(fp);
148 }

```

```

148     }
149     int main()
150     {
151         int N,f;
152         cout<<"Enter dimension of database.\n";
153         cin>>N;
154
155         base(N);
156         do
157         {
158             print(N);
159             cout << "    Menu    \n";
160             cout << "1.Count alphabets with needed characters\n";
161             cout << "2.Sort by letters number\n";
162             cout << "3.Add new element after certain\n";
163             cout << "4.Change place of two elements\n";
164             cout << "5.Save data to the file\n";
165             cout << "6.Exit\n";
166             cin >> f;
167             switch (f)
168             {
169                 case 1: {how(N); break;}
170                 case 2: { sort(N); break;}
171                 case 3: { adel(N); N++; break;}
172                 case 4: { changel(N); break;}
173                 case 5: { write(N); break;}
174                 case 6: break;
175             }
176         } while (f != 6);
177     }

```

Малюнок №1 – Контрольний приклад

```

clang version 7.0.0-3~ubuntu18.04.1 (tags/RELEASE_700/final)
❖ clang++-7 -pthread -o main main.cpp
❖ ./main
Enter dimension of database.
2
Enter letters of the alphabet
qwerty
Enter special symbols of the alphabet
1234567890
Enter letters of the alphabet
qwertyuiop
Enter special symbols of the alphabet
123456
1. Alphabet contains 6 letters: qwerty and 10 special symbols: 1234567890
2. Alphabet contains 10 letters: qwertyuiop and 6 special symbols: 123456
Menu
1.Count alphabets with needed characters
2.Sort by letters number
3.Add new element after certain
4.Change place of two elements
5.Save data to the file
6.Exit
1
Enter how much symbols you require
16
i + 1 .Alphabet contains 6 letters: qwerty and 10 special symbols : 1234567890
i + 1 .Alphabet contains 10 letters: qwertyuiop and 6 special symbols : 123456
Total 2

```

```

Enter dimension of database.
2
Enter letters of the alphabet
qwertyuiop
Enter special symbols of the alphabet
1234567890
Enter letters of the alphabet
qwerty
Enter special symbols of the alphabet
1234
1. Alphabet contains 10 letters: qwertyuiop and 10 special symbols: 1234567890
2. Alphabet contains 6 letters: qwerty and 4 special symbols: 1234
Menu
1.Count alphabets with needed characters
2.Sort by letters number
3.Add new element after certain
4.Change place of two elements
5.Save data to the file
6.Exit
2
Base is sorted
1. Alphabet contains 6 letters: qwerty and 4 special symbols: 1234
2. Alphabet contains 10 letters: qwertyuiop and 10 special symbols: 1234567890

```

```

Enter dimension of database.
2
Enter letters of the alphabet
qwerty
Enter special symbols of the alphabet
12356
Enter letters of the alphabet
qwertyuiop
Enter special symbols of the alphabet
123456
1. Alphabet contains 6 letters: qwerty and 5 special symbols: 12356
2. Alphabet contains 10 letters: qwertyuiop and 6 special symbols: 123456
Menu
1.Count alphabets with needed characters
2.Sort by letters number
3.Add new element after certain
4.Change place of two elements
5.Save data to the file
6.Exit
3
Enter number of element after which you want to add
2
Enter letters of the alphabet
asdfghjkl
Enter special symbols of the alphabet
34567890
1. Alphabet contains 6 letters: qwerty and 5 special symbols: 12356
2. Alphabet contains 10 letters: qwertyuiop and 6 special symbols: 123456
3. Alphabet contains 9 letters: asdfghjkl and 8 special symbols: 34567890
Menu
1.Count alphabets with needed characters
2.Sort by letters number
3.Add new element after certain
4.Change place of two elements
5.Save data to the file
6.Exit

```

```

1. Alphabet contains 6 letters: qwerty and 5 special symbols: 12356
2. Alphabet contains 10 letters: qwertyuiop and 6 special symbols: 123456
3. Alphabet contains 9 letters: asdfghjkl and 8 special symbols: 34567890
Menu
1.Count alphabets with needed characters
2.Sort by letters number
3.Add new element after certain
4.Change place of two elements
5.Save data to the file
6.Exit
4
sh: 1: cls: not found
Enter elements to swap
1
3
1. Alphabet contains 9 letters: asdfghjkl and 8 special symbols: 34567890
2. Alphabet contains 10 letters: qwertyuiop and 6 special symbols: 123456
3. Alphabet contains 6 letters: qwerty and 5 special symbols: 12356
Menu
1.Count alphabets with needed characters
2.Sort by letters number
3.Add new element after certain
4.Change place of two elements
5.Save data to the file
6.Exit

```

```
file.txt
1 asdfghjkl
2
3
4 qwertyuiop
5
6
7 qwertykl
8
9
```

```
1. Alphabet contains 6 letters: qwerty and 5 special symbols: 12356
2. Alphabet contains 10 letters: qwertyuiop and 6 special symbols: 1
3. Alphabet contains 9 letters: asdfghjkl and 8 special symbols: 345
Menu
1.Count alphabets with needed characters
2.Sort by letters number
3.Add new element after certain
4.Change place of two elements
5.Save data to the file
6.Exit
4
sh: 1: cls: not found
Enter elements to swap
1
3
1. Alphabet contains 9 letters: asdfghjkl and 8 special symbols: 345
2. Alphabet contains 10 letters: qwertyuiop and 6 special symbols: 1
3. Alphabet contains 6 letters: qwerty and 5 special symbols: 12356
Menu
1.Count alphabets with needed characters
2.Sort by letters number
3.Add new element after certain
4.Change place of two elements
5.Save data to the file
6.Exit
5
1. Alphabet contains 9 letters: asdfghjkl and 8 special symbols: 345
2. Alphabet contains 10 letters: qwertyuiop and 6 special symbols: 1
3. Alphabet contains 6 letters: qwerty and 5 special symbols: 12356
Menu
1.Count alphabets with needed characters
2.Sort by letters number
3.Add new element after certain
4.Change place of two elements
5.Save data to the file
6.Exit
```

Висновки. В ході виконання практичної роботи було розроблено програму для роботи з списками та збереження інформації зі списку на окремому .txt файлі у вигляді бінарного коду.