# Лабораторная работа № 6 "Строки"

1. Постановка задачи

Задана строка, состоящая из символов. Символы объединяются в слова. Слова друг от друга отделяются одним или несколькими пробелами. В конце текста ставится точка. Текст содержит не более 255 символов. Выполнить ввод строки, используя функцию Gets(s) и обработку строки в соответствии со своим вариантом.

Напечатать все слова, которые содержат по одной цифре.

2. Код программы

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

using namespace std;

int count\_digit(char\* word);

int main()

{

setlocale(LC\_ALL, "rus");

char s[256];

cout << "Введите строку: ";

gets\_s(s);

char\* word = strtok(s, " ");

while (word != NULL)

{

int count = count\_digit(word);

if (count == 1)

cout << word << endl;

word = strtok(NULL, " ");

}

}

int count\_digit(char\* word)

{

int count = 0;

for (int i = 0; i < strlen(word); i++)

{

if (isdigit(word[i]))

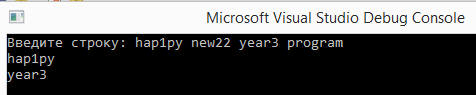
count++;

}

return count;

}

1. Результат выполнения



# Лабораторная работа № 7 7.1. "Перегрузка функций в Си++"

1. Постановка задания

Написать перегруженные функции и основную программу, которая их вызывает.

а) для вычитания целых чисел;

б) для вычитания комплексных чисел.

2. Код программы

#include <iostream>

#include <complex>

using namespace std;

int subtract(int a, int b)

{

int res = a - b;

return res;

}

complex<int> subtract(complex<int> c1, complex<int> c2)

{

complex<int> res = c1 - c2;

return res;

}

int main()

{

setlocale(LC\_ALL, "rus");

cout << "Введите первое число: ";

int num1;

cin >> num1;

cout << "Введите второе число: ";

int num2;

cin >> num2;

int res = subtract(num1, num2);

cout << "Результат вычитания = " << res << endl;

int re1, im1, re2, im2;

cout << "Введите первое комлексное число" << endl;

cout << "Re = ";

cin >> re1;

cout << "Im = ";

cin >> im1;

cout << "Введите второе комлексное число" << endl;

cout << "Re = ";

cin >> re2;

cout << "Im = ";

cin >> im2;

complex<int> c1(re1,im1);

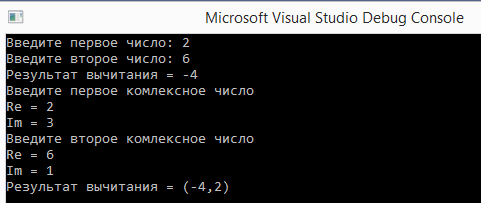
complex<int> c2(re2, im2);

complex<int> res2 = subtract(c1, c2);

cout << "Результат вычитания = " << res2 << endl;

}

3. Результат выполнения



# 7.2. "Функции с переменным числом параметров"

1. Постановка задачи

Решить указанную в варианте задачу, используя функции с переменным числом параметров.

Написать функцию sum с переменным числом параметров, которая находит сумму чисел типа int по формуле: S=a1\*a2+a3\*a4+a5\*a6+. . . . .

Написать вызывающую функцию main, которая обращается к функции sum не менее трех раз с количеством параметров 8, 10, 12.

1. Код решения

#include <iostream>

using namespace std;

int sum(int k, ...)

{

int\* p = &k;

p++;

int res = 0;

for (int i = 0; i < k; i+=2)

{

res += \*p \* \*(p + 1);

p += 2;

}

return res;

}

int main()

{

setlocale(LC\_ALL, "rus");

int sum1 = sum(8, 1, 2, 3, 4, 5, 6, 7, 8);

int sum2 = sum(10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10);

int sum3 = sum(12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12);

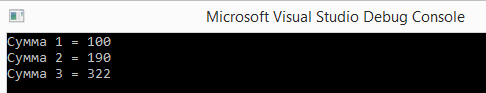
cout << "Сумма 1 = " << sum1 << endl;

cout << "Сумма 2 = " << sum2 << endl;

cout << "Сумма 3 = " << sum3 << endl;

}

1. Результат выполнения



# Лабораторная работа №8 "Блоковый ввод-вывод"

1. Постановка задачи

Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление и добавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляемых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

Структура "Человек":

* фамилия, имя, отчество;
* домашний адрес;
* номер телефона;
* возраст.

Удалить все элементы с заданным возрастом, добавить элемент после элемента с заданным номером.

1. Код решения

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <iomanip>

using namespace std;

struct Person

{

char surname[30], name[30], middlename[30];

char address[100];

char phone[20];

int age;

};

void form\_file();

void read\_file();

void work\_file();

void delete\_from\_file(int startAge, int endAge);

Person create();

int main()

{

setlocale(LC\_ALL, "rus");

int oper;

do

{

cout << "1.Сформировать файл" << endl;

cout << "2.Работа с файлом" << endl;

cout << "3.Выход" << endl;

cin >> oper;

switch (oper)

{

case 1:

form\_file();

break;

case 2:

work\_file();

break;

default:

break;

}

} while (oper != 3);

}

Person create()

{

cout << "Фамилия: ";

char surname[30];

cin >> surname;

cout << "Имя: ";

char name[30];

cin >> name;

cout << "Отчество: ";

char middlename[30];

cin >> middlename;

cout << "Возраст: ";

int age;

cin >> age;

cout << "Адрес: ";

char address[100];

cin.ignore();

gets\_s(address);

cout << "Телефон: ";

char phone[20];

cin >> phone;

Person p;

strcpy(p.address, address);

strcpy(p.middlename, middlename);

strcpy(p.name, name);

strcpy(p.phone, phone);

strcpy(p.surname, surname);

p.age = age;

return p;

}

void form\_file()

{

cout << "Кол-во людей = ";

int count;

cin >> count;

FILE\* file = fopen("file.dat", "wb");

if (file == NULL)

exit(1);

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

{

Person p = create();

fwrite(&p, sizeof(Person), 1, file);

if (ferror(file))

exit(2);

}

fclose(file);

}

void read\_file()

{

FILE\* file = fopen("file.dat", "rb");

Person p;

cout << setw(25) << "ФИО" << setw(10) << "Возраст" << setw(11) << "Телефон" << setw(20) << "Адрес" << endl;

while (fread(&p, sizeof(Person), 1, file))

{

char fio[30] = "";

strcat(fio, p.surname);

strcat(fio, " ");

strcat(fio, p.name);

strcat(fio, " ");

strcat(fio, p.middlename);

cout << setw(25) << fio << setw(10) << p.age << setw(11) << p.phone << setw(20) << p.address << endl;

}

fclose(file);

}

void delete\_from\_file(int startAge, int endAge)

{

FILE\* file = fopen("file.dat", "rb");

FILE\* tempfile = fopen("temp.dat", "wb");

Person p;

while (fread(&p, sizeof(Person), 1, file))

{

if (p.age < startAge || p.age > endAge)

{

fwrite(&p, sizeof(Person), 1, tempfile);

}

}

fclose(file);

fclose(tempfile);

remove("file.dat");

rename("temp.dat", "file.dat");

}

void add\_person(Person t, int pos)

{

if (pos < 1)

cout << "Некорректный номер" << endl;

else

{

FILE\* file = fopen("file.dat", "rb");

FILE\* tempfile = fopen("temp.dat", "wb");

Person p;

int index = 0;

while (fread(&p, sizeof(Person), 1, file))

{

fwrite(&p, sizeof(Person), 1, tempfile);

index++;

if (index == pos)

fwrite(&t, sizeof(Person), 1, tempfile);

}

fclose(file);

fclose(tempfile);

remove("file.dat");

rename("temp.dat", "file.dat");

if (index < pos)

cout << "Некорректный номер" << endl;

}

}

void work\_file()

{

int oper;

do

{

cout << "1.Чтение файла" << endl;

cout << "2.Удаление из файла" << endl;

cout << "3.Добавление в файл" << endl;

cout << "0.Назад" << endl;

cin >> oper;

switch (oper)

{

case 1:

read\_file();

break;

case 2:

{

int start, end;

cout << "Введете начальный возраст: ";

cin >> start;

cout << "Введете конечный возраст: ";

cin >> end;

delete\_from\_file(start, end);

break;

}

case 3:

{

int pos;

cout << "После какого элемента добавить? ";

cin >> pos;

Person p = create();

add\_person(p, pos);

break;

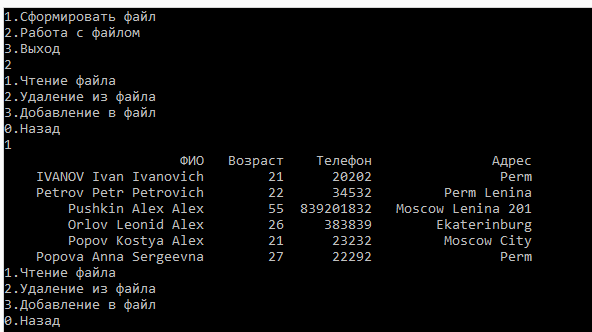
}

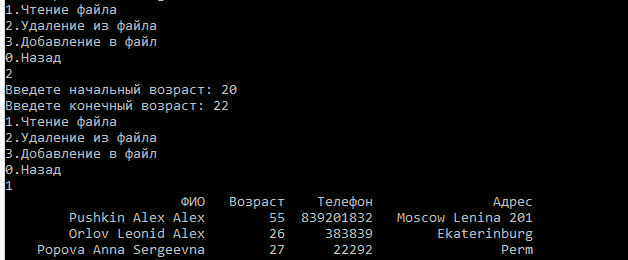
}

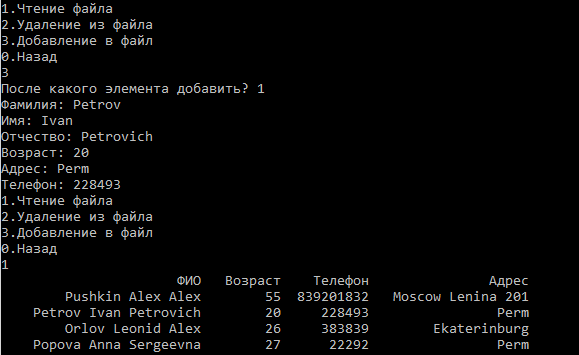
} while (oper != 0);

}

1. Результат выполнения







# Лабораторная работа № 9 "Строковый ввод-вывод"

1. Постановка задачи

* Создать текстовый файл F1 не менее, чем из 10 строк и записать в него информацию
* Выполнить задание.

1. Скопировать из файла F1 в файл F2 строки, начиная с 4.
2. Подсчитать количество символов в последнем слове F2.
3. Код программы

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

using namespace std;

void copy\_file(const char\* source, const char\* destination, int pos);

void calc\_count\_symbols(const char\* source);

int main()

{

setlocale(LC\_ALL, "rus");

copy\_file("file.txt", "out.txt", 4);

calc\_count\_symbols("out.txt");

}

void copy\_file(const char\* source, const char\* destination, int pos)

{

FILE\* file = fopen(source, "r");

FILE\* outfile = fopen(destination, "w");

char s[255];

int numberLine = 0;

while (!feof(file))

{

fgets(s, 255, file);

numberLine++;

if (numberLine >= pos)

{

fputs(s, outfile);

}

}

fclose(file);

fclose(outfile);

}

void calc\_count\_symbols(const char\* source)

{

FILE\* file = fopen(source, "r");

char s[255];

while (!feof(file))

fgets(s, 255, file);

char lastRow[255] = "";

strcpy(lastRow, s);

char\* word = strtok(s, " ,.!?");

char lastWord[100];

while (word != NULL)

{

strcpy(lastWord, word);

word = strtok(NULL, " ,.!?");

}

cout << "Последняя строка: " << lastRow << endl;

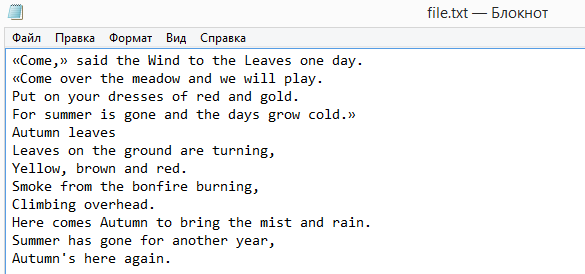
cout << "Последнее слово: " << lastWord << endl;

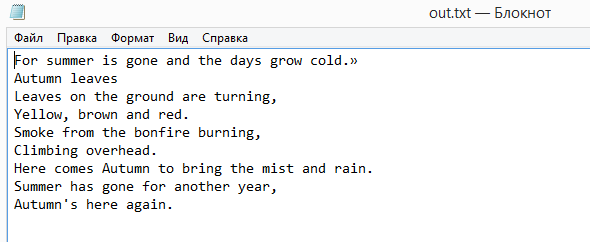
cout << "Кол-во символов в последнем слове = " << strlen(lastWord) << endl;

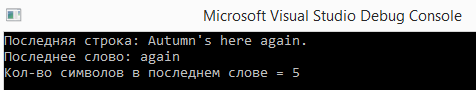
fclose(file);

}

1. Результат выполнения







# Лабораторная работа № 10 "Динамические массивы"

1. Постановка задачи

Написать программу, в которой создаются динамические массивы и выполнить их обработку в соответствии со своим вариантом.

Сформировать одномерный массив. Удалить из него элемент с заданным номером, добавить К элементов, начиная с заданного номера;

1. Код программы

#include <iostream>

#include <ctime>

using namespace std;

void fill\_array(int\* a, int n)

{

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

{

a[i] = rand() % 100;

}

}

void print\_array(int\* a, int n)

{

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

{

cout << a[i] << " ";

}

cout << endl;

}

bool delete\_element(int\* a, int& n, int num\_del)

{

if (num\_del < 0 || num\_del >= n)

return false;

for (int i = num\_del; i < n-1; i++)

{

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

}

n--;

return true;

}

int\* insert\_elements(int\* a, int& n, int pos, int k)

{

if (pos >= 0 && pos <= n)

{

a = (int\*)realloc(a, (n + k) \* sizeof(int));

for (int i = n-1; i >= pos ; i--)

{

a[i + k] = a[i];

}

for (int i = pos; i < pos + k; i++)

{

a[i] = rand() % 100;

}

}

n = n + k;

return a;

}

int main()

{

setlocale(LC\_ALL, "rus");

srand(time(0));

cout << "n = ";

int n;

cin >> n;

int\* a = (int\*)malloc(n \* sizeof(int));

fill\_array(a, n);

print\_array(a, n);

int num\_del;

cout << "Номер удаляемого элемента: ";

cin >> num\_del;

bool res = delete\_element(a, n, num\_del);

if (res == true)

print\_array(a, n);

else

cout << "Некорректный номер удаляемого элемента" << endl;

cout << "Куда добавить элементы? ";

int pos;

cin >> pos;

cout << "Сколько эл-тов добавить? ";

int k;

cin >> k;

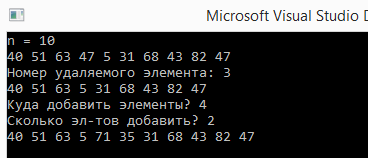
a = insert\_elements(a, n, pos, k);

print\_array(a, n);

free(a);

}

1. Результат выполнения



# Лабораторная работа № 11 "Информационные динамические структуры"

1. Постановка задачи

Написать программу, в которой создаются динамические структуры и выполнить их обработку в соответствии со своим вариантом.

Для каждого вариант разработать следующие функции:

1. Создание списка.
2. Добавление элемента в список (в соответствии со своим вариантом).
3. Удаление элемента из списка (в соответствии со своим вариантом).
4. Печать списка.
5. Запись списка в файл.
6. Уничтожение списка.
7. Восстановление списка из файла.

Записи в линейном списке содержат ключевое поле типа int. Сформировать однонаправленный список. Удалить из него элемент с заданным номером, добавить К элементов, начиная с заданного номера;

1. Код программы

#include <iostream>

#include <fstream>

using namespace std;

struct point

{

int key;

point\* next;

};

point\* make\_point(int n)

{

point\* first, \* p;

first = NULL;

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

{

p = new point();

cout << "Введите значение: ";

cin >> p->key;

p->next = first;

first = p;

}

return first;

}

void print(point\* first)

{

if (first == NULL)

cout << "Список пустой" << endl;

else

{

point\* p = first;

while (p != NULL)

{

cout << p->key << " ";

p = p->next;

}

cout << endl;

}

}

point\* get\_last(point\* first)

{

while (first->next != NULL)

{

first = first->next;

}

return first;

}

point\* delete\_element(point\* first, int number)

{

if (first == NULL)

{

cout << "Список пустой" << endl;

return first;

}

if (number < 1)

{

cout << "Неверный номер элемента" << endl;

return first;

}

if (number == 1)

{

point\* temp = first;

first = first->next;

delete temp;

}

else

{

int i = 1;

point\* p = first;

while (p != NULL && i < number - 1)

{

i++;

p = p->next;

}

if (p == NULL)

cout << "Неверный номер элемента" << endl;

else

{

point\* temp = p->next;

p->next = p->next->next;

delete temp;

}

}

return first;

}

point\* add\_elements(point\* first, int number, int k)

{

if (first == NULL)

{

cout << "Список пустой" << endl;

return first;

}

if (number < 1)

{

cout << "Неверный номер элемента" << endl;

return first;

}

if (number == 1)

{

point\* beg\_new\_list = make\_point(k);

point\* end\_new\_list = get\_last(beg\_new\_list);

end\_new\_list->next = first;

first = beg\_new\_list;

}

else

{

int i = 1;

point\* p = first;

while (p != NULL && i < number - 1)

{

i++;

p = p->next;

}

if (p == NULL)

cout << "Неверный номер элемента" << endl;

else

{

point\* beg\_new\_list = make\_point(k);

point\* end\_new\_list = get\_last(beg\_new\_list);

end\_new\_list->next = p->next;

p->next = beg\_new\_list;

}

}

return first;

}

void free\_list(point\*\* first)

{

point\* p = \*first, \* t;

while (p != NULL)

{

t = p->next;

delete p;

p = t;

}

\*first = NULL;

}

void save\_to\_file(point\* first, const char\* filename)

{

if (first == NULL)

{

cout << "Список пустой" << endl;

return;

}

ofstream file(filename);

point\* p = first;

while (p != NULL)

{

file << p->key;

if (p->next != NULL)

file << " ";

p = p->next;

}

file.close();

}

point\* read\_file(const char\* filename)

{

ifstream file(filename);

if (!file.is\_open())

{

cout << "Не удалось открыть файл" << endl;

return NULL;

}

point\* beg = NULL, \* p, \* pred = NULL;

int num;

while (!file.eof())

{

file >> num;

p = new point();

p->key = num;

p->next = NULL;

if (beg == NULL)

beg = p;

else

pred->next = p;

pred = p;

}

file.close();

return beg;

}

int main()

{

setlocale(LC\_ALL, "rus");

point\* beg = NULL;

int oper;

do

{

system("cls");

cout << "1.Создать список" << endl;

cout << "2.Удалить элемент с заданным номером" << endl;

cout << "3.Вставка элементов" << endl;

cout << "4.Вывод списка" << endl;

cout << "5.Сохранение в файл" << endl;

cout << "6.Загрузка из файла" << endl;

cout << "0.Выход" << endl;

cin >> oper;

switch (oper)

{

case 1:

{

int n;

cout << "Кол-во эл-тов списка = ";

cin >> n;

beg = make\_point(n);

print(beg);

break;

}

case 2:

{

print(beg);

int number;

cout << "Введите номер: ";

cin >> number;

beg = delete\_element(beg, number);

print(beg);

break;

}

case 3:

{

print(beg);

int number;

cout << "Введите номер: ";

cin >> number;

int k;

cout << "Сколько элементов добавить: ";

cin >> k;

beg = add\_elements(beg, number, k);

print(beg);

break;

}

case 4:

{

print(beg);

break;

}

case 5:

{

char filename[200];

cout << "Введите название файла: ";

cin >> filename;

save\_to\_file(beg, filename);

free\_list(&beg);

print(beg);

break;

}

case 6:

{

char filename[200];

cout << "Введите название файла: ";

cin >> filename;

beg = read\_file(filename);

print(beg);

break;

}

default:

break;

}

if (oper != 0)

system("pause");

} while (oper != 0);

free\_list(&beg);

}

1. Результат выполнения

