Цель:

Изучить основные принципы написания и объявления функций, ознакомиться с возможностями передачи параметров в функции и получения значения, возвращаемого функцией.

**Вариант 3.** Задание 1

Текст задания

Описать процедуру NMinmax(A, N, NMin, NMax), находящую номера минимального и максимального элемента вещественного массива A размера N. Выходные параметры целого типа: NMin (номер минимального элемента) и NMax (номер максимального элемента). С помощью этой процедуры найти номера минимальных и максимальных элементов массивов A, B, C размера NA, NB, NC соответственно.

Описание входных и выходных данных

Входными данными являются количества элементов (целый тип) NA, NB, NC массивов A, B, C и их элементы вещественного типа. Выходными данными является номера (целый тип) минимальных и максимальных элементов массивов A, B, C.

Блок-схема алгоритма

Текст программы

#include <stdio.h>

void NMinmax(float\* A, int\* N, int& NMin, int& NMax);

int main() {

int NA=0, NB=0, NC=0, NMin, NMax;

int\* arrOfNs[3] = {&NA, &NB, &NC};

scanf("%d %d %d", &NA, &NB, &NC);

float A[NA];

float B[NB];

float C[NC];

float\* arrOfArrs[3] = {A, B, C};

for (int j=0; j<3; j++) {

for (int i=0; i<\*(arrOfNs[j]); i++) scanf("%f", &(arrOfArrs[j][i]));

}

for (int i=0; i<3; i++) {

NMinmax(arrOfArrs[i], arrOfNs[i], NMin, NMax);

printf("%d %d\n", NMin+1, NMax+1);

}

return 0;

}

void NMinmax(float\* A, int\* N, int& NMin, int& NMax) {

NMin=0;

NMax=0;

for (int i=0; i<\*N; i++) {

if (A[NMin] > A[i]) NMin=i;

if (A[NMax] < A[i]) NMax=i;

}

}

Результат работы

3 3 3

1 2 3

3 2 1

3 1 2

1 3

3 1

2 1

Проверка результата работы

1. A[1] = 1 < A[3] = 3
2. B[3] = 1 < B[1] = 3
3. C[2] = 1 < C[1] = 3

**Вариант 1.** Задание 2

Текст задания

Описать функцию IsIdent(S) целого типа, проверяющую, является ли строка S допустимым идентификатором, то есть непустой строкой, которая содержит только латинские буквы, цифры и символ подчеркивания «\_» и не начинается с цифры. Если S является допустимым идентификатором, то функция возвращает 0. Если S является пустой строкой, то возвращается –1, если S начинается с цифры, то возвращается –2. Если S содержит недопустимые символы, то возвращается номер первого недопустимого символа. Проверить с помощью функции IsIdent пять данных строк.

Описание входных и выходных данных

Входными данными являются 5 строк символов. Выходными данными является 5 целых чисел – коды возврата функции.

Блок-схема алгоритма



Текст программы

#include <stdio.h>

int IsIndent(char\* s);

int main() {

const int MAX\_LENGTH = 10000;

int ArrayOfCodes[5];

char str[MAX\_LENGTH];

for (int i=0; i<5; i++) {

gets(str);

ArrayOfCodes[i] = IsIndent(str);

}

for (int i=0; i<5; i++) printf("%d\n", ArrayOfCodes[i]);

return 0;

}

int IsIndent(char\* s) {

if (s[0] == '\0') return -1;

if (s[0] >= '0' && s[0] <= '9') return -2;

for (int i=0; s[i]!='\0'; i++) {

if ((s[i] >= '0' && s[i] <= '9') ||

(s[i] >= 'A' && s[i] <= 'Z') ||

(s[i] >= 'a' && s[i] <= 'z') ||

(s[i] == '\_')) continue;

return i+1;

}

return 0;

}

Результат работы

qwerty+

qwerty

1qwe

Q1we\_

7

0

-2

0

-1

Проверка результата работы

На 7-ой позиции + – недопустимое значение

qwerty – латинские, всё в порядке

на 1-ой позиции цифра

Q1we\_ – допустимые символы

‘‘ – пустая строка

Вывод

В ходе лабораторной работы были реализованы пользовательские функции с и без возвращаемого значения, с передачей указателей и ссылок в качестве параметров.