

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Московский государственный технический
университет имени Н. Э. Баумана (национальный исследовательский
университет)

Факультет «Робототехника и комплексная автоматизация»
Кафедра «Системы автоматизированного проектирования»

Отчет по ознакомительной практике

По курсу «Объектно-ориентированное программирование»

Выполнил:

Студент Петраков С.А.
Группа РК6-26Б

Проверил:

Дата _____
Подпись _____

Москва, 2020 г.

Билет Т16

Задание:

Разработать программу поиска наибольшего натурального числа в заданной числовой последовательности, элементами которой могут быть любые целые и вещественные числа в неограниченном диапазоне. Последовательность чисел должна передаваться программе через поток стандартного ввода (stdin), где записи чисел разделены промежутками из символов пробела, табуляции и перевода строки. При обработке заданной последовательности программа должна игнорировать любую нецифровую информацию из потока стандартного ввода, кроме символов указанных промежутков между числами. Результат поиска должен отображаться информационном сообщении в потоке протокола стандартной диагностики (stderr) после получения программой признака конца потока стандартного ввода. В информационном сообщении должна быть указана длины последовательности, количество обнаруженных натуральных чисел и максимальное из них.

Алгоритм:

Для определения максимального натурального числа из вводимой последовательности необходимо вычленить отдельные слова (числа). Ищем последовательность, разделенную символами пробела, табуляции или перевода на следующую строку. Проверяем последовательность на то, что она является числом и что это число натуральное, и ищем максимум из всех чисел.

Алгоритм проверки числа на натуральное: проверяем, что число не 0 и что присутствуют только цифры в записи числа

Алгоритм поиска максимума: сравниваем длины строк; если строки имеют одинаковую длину сравниваем поразрядно.

Входные данные:

Последовательность символов, разделенная символами пробела, табуляции или перевода на новую строку.

Выходные данные:

Длина последовательности, количество обнаруженных натуральных чисел и максимальное из них

Текст программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int isNatural(char* str)//return 1 if Natural
{
    if (strcmp("1", str) == 1)
        return 0;
    for (int j = 0; j < (int)strlen(str); j++)
        if (!(str[j] >= '0' && str[j] <= '9'))
            return 0;
    return 1;
}

_Bool Compare(char* first, char* second)//return 1: f>s; return 0 f<=s
```

```

{
    if (strlen(first) == strlen(second))
    {
        for (int i = 0; i < (int)strlen(first); i++)
            if (first[i] > second[i])
                return 1;
    }
    else if (strlen(first) > strlen(second))
        return 1;
    return 0;
}

int main() {

    //Start Initializations
    int n = 0; //Amount of Naturals numbers
    int len = 0; //Amount of symbols

                                //Initialization temp input
    int j = 0;
    char ch = 'a';
    char* word;
    int wordlen = 32;
    if ((word = (char*)malloc(sizeof(char) * (wordlen + 1))) == NULL)
        return 1;
    //Initialize max
    char* max;
    if ((max = (char*)malloc(sizeof(char))) == NULL)
        return 1;
    max[0] = '\0';
    //End Initializations

    //Main Algorithm
    do
    {
        ch = getchar();
        len++;
        if ((ch == ' ' || ch == '\n' || ch == '\t' || ch == EOF) && (j != 0))
        {
            word[j] = '\0';
            j = 0;
            if (isNatural(word))
            {
                n++; //Count Naturals
                if (Compare(word, max)) //if(argv[i]>max)
                {
                    //start max=argv[i]
                    free(max);
                    if ((max = (char*)malloc(strlen(word) * sizeof(char) +
1)) == NULL)
                        return 1;
                    strcpy(max, word);
                    //end
                }
            }
        }
        else if (!(ch == ' ' || ch == '\n' || ch == '\t' || ch == EOF))
        {
            if (j == wordlen)
            {
                wordlen += 32;
                char* temp;
                if ((temp = (char*)malloc(sizeof(char) * (wordlen + 1))) ==
NULL)
                    return 1;
                strcpy(temp, word);
            }
        }
    } while (ch != EOF);

    printf("Number of natural numbers: %d\n", n);
    printf("Maximum value: %s\n", max);
    return 0;
}

```

```

        free(word);
        word = temp;

    }
    word[j] = ch;
    j++;
}
} while (ch != EOF);
len--;
//stderr
fprintf(stderr, "All length: %d.\nNaturals: %d.\nMax: %s.\n", len, n, max);
free(word);
return 0;
}

```

Тесты:

Номер	Вход	Выход
1	ert 3yt4	All length: 10. Naturals: 0. Max: .
2	23 45.6 r 0	All length: 12. Naturals: 1. Max: 23.

Список использованной литературы:

- Волосатова Т.М., Родионов С.В. Лекции по курсу «Объектно-ориентированное программирование»
- bigor.bmstu.ru