**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

Московский институт электроники и математики Национального

исследовательского университета "Высшая школа экономики"

Департамент прикладной математики

**ОТЧЕТ**

**По лабораторной работе №А3**

**По курсу «Компьютерный практикум»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ФИО студента | | Номер группы | Дата |
| Кармаев Александр Андреевич | БПМ211 | 24.02.2022 |
|  |
|  |
|  |

**Москва – 2022 г.**

1. **ЗАДАНИЕ А3**

Дана строка из трёх десятичных цифр. Если цифры составляют убывающую арифметическую прогрессию, то записать их в обратном порядке, иначе – умножить на 3 вторую цифру (если это возможно).

Изображение выглядит как текст

Автоматически созданное описание

1. **РЕШЕНИЕ**

#include <stdio.h>

int main(void) {

char str[4]; // массив символов - три наших числа - байты

printf("Input string with 3 numbers: ");

place:;

fgets(str, 4, stdin); // применяем fgets вместо scanf, чтобы ограничить ввод строки до 3 символов

// проверяем, что введена строка из числовых символов

if (!((int)str[0] >= 48 && (int)str[0] <= 57 && (int)str[1] >= 48 && (int)str[1] <= 57 && (int)str[2] >= 48 && (int)str[2] <= 57)) {

printf("Wrong input, try again: ");

if(!(str[0] == '\n' || str[1] == '\n' || str[2] == '\n')) // проверка, на ввод количества символов меньше, чем 3

while ((getchar()) != '\n'); // избавляемся от символов '\n' попавших в буфер

goto place;

}

\_\_asm {

; считываем введённые значения в регистры - в регистры будут записаны числовые коды символов

mov al, str[0] ; перавя цифра

mov ah, str[1] ; вторая цифра

mov bl, str[2] ; третья цифра

; проверяем, что последовательность убывает для первой пары

cmp al, ah ; устанавливаем флаги по результату : AL – AH

jnge condition ; если значения флага соответствует отношению AL < AH, то переходим к метке (condition)

jz condition; если флаги установлены на результат AL – AH = 0, то переходим на метку(condition)

; сравниваем разницу между парами соседних чисел, чтобы установить арифметическую прогрессию

sub al, ah ; в AL находится разница первого и второго числа последовательности

sub ah, bl ; в AH находится разница второго и третьего числа последовательности

cmp al, ah ; устанавливаем флаги по результату : AL – AH

jz yes ; если флаги установлены на результат AL – AH = 0, то переходим на метку (yes)

jmp condition ; если мы не перешли к метке (yes), то попадём в метку (condition)

yes:

; мы установили, что последовательность является убывающей арифметической прогрессией

; кладём ответы в исходную строку, для этого заново записываем числа последовательности в регистры, чтобы произвести перезапись

mov al, str[0] ; перавя цифра

mov bl, str[2] ; третья цифра

mov str[0], bl ; перавя цифра меняется с третьей

mov str[2], al ; третья цифра меняется с первой

jmp end ; переходим к метке (end), чтобы не зайти в код после метки (condition)

condition:

; мы установили, что последовательность не является убывающей арифметической прогрессией

; первая и третья цифра не меняется, проверяем возможность увеличить 2-ое число в 3 раза

mov al, str[1] ; вторая цифра

sub al, 30h ; вычитаем код нуля из регистра, чтобы явно работать с нашим числом, а не кодом ascii(потом надо будет сделать обратную операцию)

cmp al, 4 ; устанавливаем флаги по результату : AL – 4

jge end ; если значения флага соответствует отношению AL >= 4, то переходим к метке(end)

mov bl, 3 ; записываем число 3 в регистр, чтобы кооректоно использовать оператор умножения

imul bl ; если мы не перешли к метке(end), то умножаем число на 3

add al, 30h ; возвращаем число к символьному представлению

mov str[1], al ; кладём ответ в исходную строку

end:

}

printf("\nResult: %s\n", str);

}

**Ассемблерная вставка из Visual Studio**

**Изображение выглядит как текст

Автоматически созданное описание**

**Тесты:**

**1 – Введена убывающая арифметическая прогрессия**

**Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание**

**2 – Введена возрастающая арифметическая прогрессия**

**Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание**

**3 – Введена произвольная последовательность**

**Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание**

**4 – Обработка исключений**

**Изображение выглядит как текст

Автоматически созданное описание**

**// если вводится больше 3 символов, то лишние символы игнорируются**

**Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание**