|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
| Лабораторная работа № 1 | | |
| по дисциплине «Языки программирования и методы трансляции» | | |
| **Цепочечные команды** | | |
|  | | |
|  | Бригада | Голубь Андрей |
|  | Буданцев Дмитрий |
| Группа ПМ-13 |  |
| Вариант 3 |  |
|  |  |
|  |  |
| Преподаватель | Дворецкая Виктория Константиновна |
|  |  |
| Новосибирск,2024 | | |

1. **Постановка задачи**

Написать программу выделения из исходной строки подстроки символов заданной длины с указанного номера позиции.

1. **Алгоритм**
   * + 1. Перекодировать все строковые константы с помощью функции CharToOem
       2. Получим необходимые дескрипторы ввода-вывода и сохраним их в переменные DIN и DOUT соответственно
       3. Получим длину подстроки и номер позиции с помощью функции ReadConsole. Каждое число проверим, подходит ли оно условию с помощью подпрограммы CHECKNUM, если условие не выполняется, выходим из программы с кодом 1, иначе перекодируем два числа из строчного формата в числовой с помощью подпрограммы CONVERTCHARTOINT.
       4. Получаем исходную строчку с помощью подпрограммы ReadConsole, после проверяем, что длина исходной строки не меньше суммы введенных в шаге 3 переменных с помощью подпрограммы CHECKSHAPE, если условие не выполняется, выходим из программы с кодом 1.
       5. Получаем подстроку из исходной строчки, которая была получена на шаге 4, с указанного номера и определённого размера, используя данные которые были получены на шаге 3, с помощью подпрограммы SUBSTRING
       6. Выводим результат шага 5.
       7. Завершаем программу с кодом 0.
2. **Результаты тестирования программы**

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Выходные данные** | **Цель теста** |
| 10  3  ->>Результат! | Результат! | Проверка корректности работы, простой тест |
| 4  49  Я мог с лёгкостью получить красный диплом, но не стал, ведь на красный идти нельзя | стал | Использование строки имеющую большую длинну |
| 0  2  Пара – это не те, что живут вместе. Пара – это то на что я не хожу |  | Нулевая длина |
| -228  4  Не ругайте ленивых они же ничего не сделали | ERROR | Использование чисел не подходящие под условие |

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Выходные данные** | **Цель теста** |
| 9  -1337  Умные мысли часто преследуют меня, но я быстрее | ERROR | Использование чисел не подходящие под условие |
| 1234  54  Без шуток | ERROR |

1. **Текст программы**

.386

.MODEL FLAT, STDCALL

OPTION CASEMAP: NONE

EXTERN  GetStdHandle@4: PROC

EXTERN  WriteConsoleA@20: PROC

EXTERN  CharToOemA@8: PROC

EXTERN  ReadConsoleA@20: PROC

EXTERN  ExitProcess@4: PROC

EXTERN  lstrlenA@4: PROC

.DATA

STRN1 db "Введите строчку : ",13,10,0

STRN2 db "Введите длину подстроки: ", 13, 10, 0;

STRN3 db "Введите номер позиции: ", 13, 10, 0

DERR db "ERROR: Данные, введены некорректно.", 13, 10, 0;

DIN DD ?

DOUT DD ?

BUF  DB 200 dup (?)

NEW\_STRING DB 200 dup (?)

LENS DD ?

LENS\_SUBSTRING DD ?

START\_STRING DD ?

.CODE

CONVERTCHARTOINT PROC

MOV ESI, OFFSET BUF

MOV ECX, LENS

XOR EBX, EBX

MOV EAX, 0

MOV BL, [ESI];

CMP BL, '-'

JNE NO\_Otr

JE Otr

NO\_Otr:

    CONVERT:

        MOV EDX, 10

        MOV BL, [ESI];

        SUB BL, '0';

        MUL EDX;

        ADD EAX, EBX

        INC ESI;

    LOOP CONVERT;

    jmp end\_proces

Otr:

    INC ESI;

    CONVERT\_1:

        MOV EDX, 10

        MOV BL, [ESI];

        SUB BL, '0';

        IMUL EDX;

        ADD EAX, EBX

        INC ESI;

    LOOP CONVERT\_1;

    MOV EDX, -1

    IMUL EDX

end\_proces:

    ret

CONVERTCHARTOINT ENDP

CHECKNUM PROC

MOV ESI, OFFSET BUF

MOV BL, [ESI];

CMP BL, '-'

JE Error

JNE Not\_err

Error:

    PUSH OFFSET DERR

    CALL lstrlenA@4

    PUSH 0

    PUSH OFFSET LENS

    PUSH EAX

    PUSH OFFSET DERR

    PUSH DOUT

    CALL WriteConsoleA@20

    PUSH 1

    CALL ExitProcess@4

Not\_err:

    ret

CHECKNUM ENDP

CHECKSHAPE PROC

MOV EAX, LENS\_SUBSTRING

ADD EAX, START\_STRING

CMP LENS, EAX

JAE Correct

JB Error

Error:

    PUSH OFFSET DERR

    CALL lstrlenA@4

    PUSH 0

    PUSH OFFSET LENS

    PUSH EAX

    PUSH OFFSET DERR

    PUSH DOUT

    CALL WriteConsoleA@20

    PUSH 1

    CALL ExitProcess@4

Correct:

    ret

CHECKSHAPE ENDP

SUBSTRING PROC

CLD

LEA ESI, BUF

LEA EDI, NEW\_STRING

ADD ESI, START\_STRING

MOV ECX, LENS\_SUBSTRING

REP MOVSB

ret

SUBSTRING ENDP

MAIN PROC

MOV  EAX, OFFSET STRN1

PUSH EAX

PUSH EAX

CALL CharToOemA@8;

MOV  EAX, OFFSET STRN2

PUSH EAX

PUSH EAX

CALL CharToOemA@8

MOV  EAX, OFFSET STRN3

PUSH EAX

PUSH EAX

CALL CharToOemA@8

MOV  EAX, OFFSET DERR

PUSH EAX

PUSH EAX

CALL CharToOemA@8

PUSH -10

CALL GetStdHandle@4

MOV DIN, EAX

PUSH -11

CALL GetStdHandle@4

MOV DOUT, EAX

PUSH OFFSET STRN2

CALL lstrlenA@4

PUSH 0

PUSH OFFSET LENS

PUSH EAX

PUSH OFFSET STRN2

PUSH DOUT

CALL WriteConsoleA@20

PUSH 0

PUSH OFFSET LENS

PUSH 200

PUSH OFFSET BUF

PUSH DIN

CALL ReadConsoleA@20

SUB LENS, 2;

CALL CHECKNUM;

CALL CONVERTCHARTOINT;

MOV LENS\_SUBSTRING, EAX

PUSH OFFSET STRN3

CALL lstrlenA@4

PUSH 0

PUSH OFFSET LENS

PUSH EAX

PUSH OFFSET STRN3

PUSH DOUT

CALL WriteConsoleA@20

PUSH 0

PUSH OFFSET LENS

PUSH 200

PUSH OFFSET BUF

PUSH DIN

CALL ReadConsoleA@20

SUB LENS, 2;

CALL CHECKNUM;

CALL CONVERTCHARTOINT;

MOV START\_STRING, EAX

PUSH OFFSET STRN1

CALL lstrlenA@4

PUSH 0

PUSH OFFSET LENS

PUSH EAX

PUSH OFFSET STRN1

PUSH DOUT

CALL WriteConsoleA@20

PUSH 0

PUSH OFFSET LENS

PUSH 200

PUSH OFFSET BUF

PUSH DIN

CALL ReadConsoleA@20

SUB LENS, 2

CALL CHECKSHAPE

CALL SUBSTRING

PUSH 0

PUSH OFFSET LENS

PUSH LENS\_SUBSTRING

PUSH OFFSET NEW\_STRING

PUSH DOUT

CALL WriteConsoleA@20

PUSH 0

CALL ExitProcess@4

MAIN ENDP

END MAIN