Косинов Денис Константинович БПИ239.

Вариант 12.

АВС ИДЗ#3.

**Задача:**

Разработать программу, определяющую минимальный и максимальный (по числовому значению) символ в заданной ASCII–строке. Вывести в файл символ и его числовой код.

**Код программы:**

https://github.com/o7Techno/CSA-HW/tree/main/%D0%90%D0%92%D0%A1%20%D0%98%D0%94%D0%97%233

**Описание программы:**

Программа разбита на 3 файла: IDZ.ams выступает main файлом, там запускается сама программа и оттуда идут вызовы дальнейших функций; IDZ\_functions.asm содержит в себе все подпрограммы вызываемые в программе; macroslib.inc - библиотека макро.

**Описание подпрограмм (более подробное объяснения для передачи параметров в подпрограммы можно найти в комментариях в коде):**

read\_string - читает строку с клавиатуры.

print\_string - выводит на дисплей строковое значение, адрес которого находится в регистре a0.

read\_int - считывает int значение с клавиатуры и возвращает его в регистр a0.

print\_int - выводит int значение на дисплей.

read\_char - читает char с клавиатуры.

print\_char - выводит char на дисплей.

open - открывает файл.

close - закрывает файл.

read\_file - читает данные с файла.

write\_file - записывает данные в файл.

min\_byte - рассчитывает минимальный по значению байт в строке.

max\_byte - рассчитывает максимальный по значению байт в строке.

**Описание макро (доп. информация также присутствует в комментариях к коду):**

int\_input\_macro (%register) - считывает int значение с клавиатуры при помощи подпрограммы read\_int и записывает его в регистр %register (Обертка над read\_int).

int\_output\_macro (%register) - записывает int значение с консоли в регистр %register при помощи write\_int.

string\_input\_macro(%buffer\_address %max\_buffer\_size) - inputs a string value and stores it in buffer from address %buffer\_address with size %max\_buffer\_size.

string\_output\_macro (%address) - выводит на дисплей string значение, находящееся по адресу %address при помощи подпрограммы print\_string (Обертка над print\_string).

char\_input\_macro(%register) - inputs a char and stores it in register %register.

char\_output\_macro(%register) - outputs a char in console from register %register.

remove\_n (%address) - removes \n left at the end of every string input from console in RARS.

clear (%address) - clears 16 bytes from address %address.

max\_byte\_macro(%address) - calculates the maximal byte in a string given at address %address using max\_byte subroutine.

min\_byte\_macro(%address) - calculates the minimal byte in a string given at address %address using min\_byte subroutine.

open\_macro(%file\_name, %opt) - opens a file with name %file\_name and options %opt using open subroutine.

close\_macro(%file\_descriptor) - closes a file with file descriptor %file\_descriptor using close subroutine.

file\_read\_macro(%file\_descriptor, %reg, %size) - reads data from file using read\_file subroutine.

file\_write\_macro(%file\_descriptor, %buffer, %size) - writes data into file using write\_file subroutine.

convert\_macro(%address) - converts an integer from address into string.

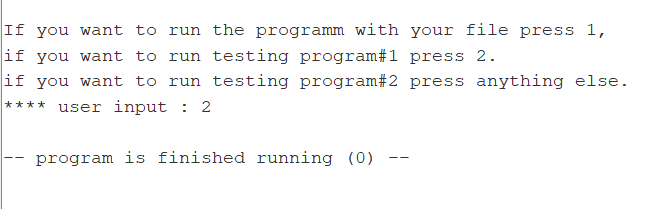
allocate(%size) - allocates memory of size %size in heap.

exit - exits the program.

**Как использовать программу:**

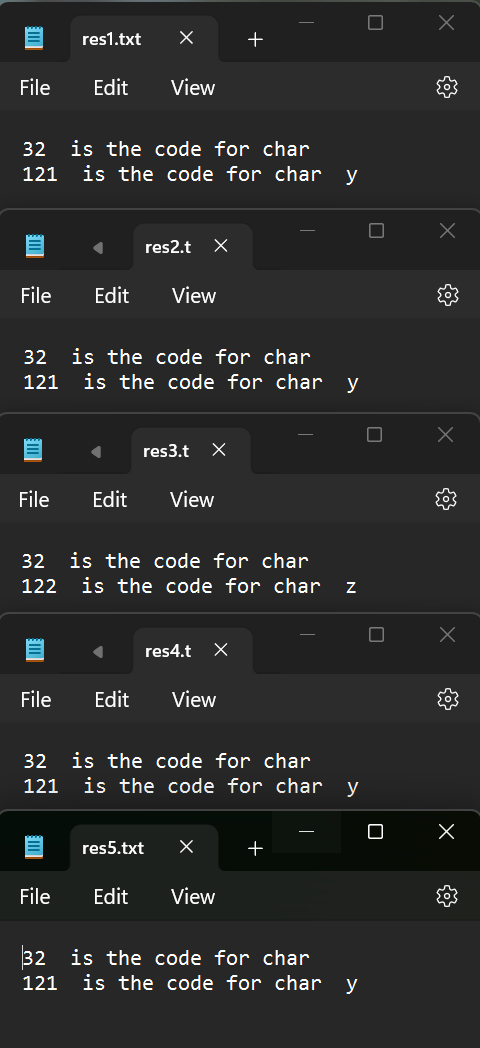
Первый ввод пользователя - режим, в котором он хочет использовать данную программу. (1 - ввод данных самому, 2 - автоматическая тестовая программа#1, любое другое - автоматическая тестовая программа#2). Далее, если пользователь ввел не 1, то программа создает файлы результаты прогона по тестовым файлам и на этом программа завершается, если же пользователь ввел 1 на выборе режима, то следующий его ввод будет означать названия файла для обработки. Затем пользователь вводит имя файла для записи данных. После чего пользователю предлагается вывести результат в консоль, если пользователь нажимает Y, то результат выводится в консоль и программа завершается; N - программа завершается.

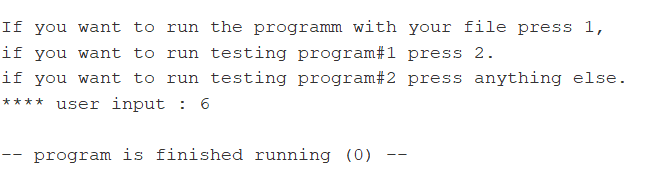
**Тесты:**

****

Автоматическое тестирование#1.

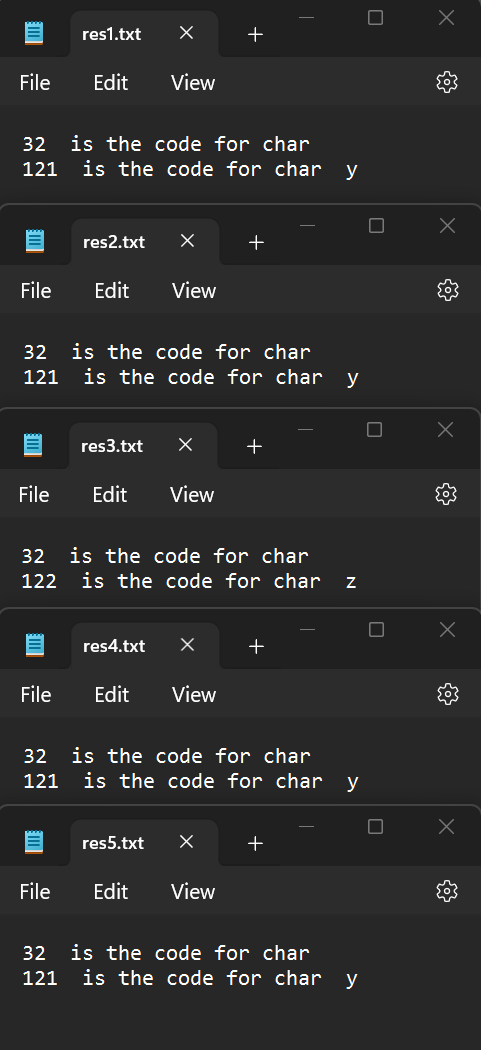
Файлы которое оно сгенерировало:





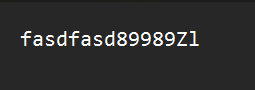
Автоматическое тестирование#2.

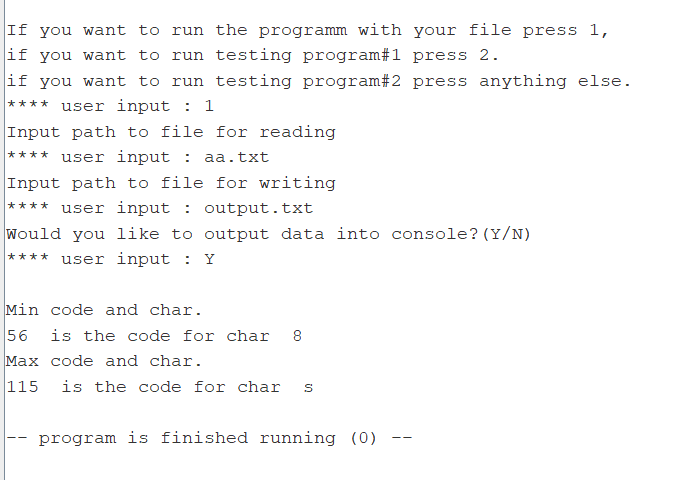
Файлы которое оно сгенерировало:



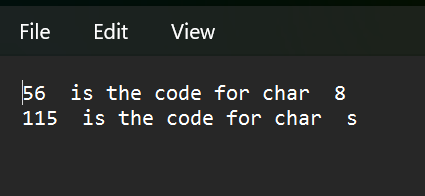
Произвольные данные:

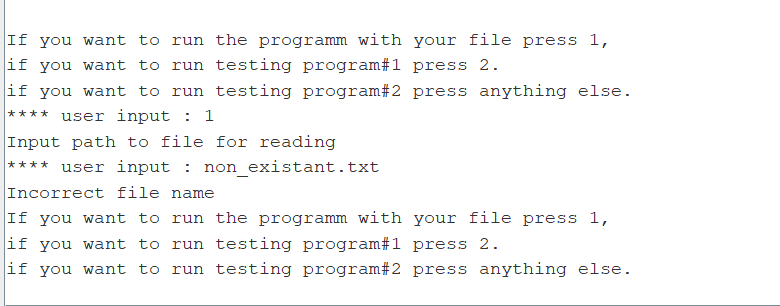
Входной файл:





Выходной файл:





Некорректное название файла для ввода.

**На 4-5 баллов:**

Есть решение задачи, комментарии присутствуют, тесты можно увидеть выше, обработка данных, полученных из файла сформирована в виде отдельной подпрограммы, файлы для тестов присутствуют, все изменяемые параметры программы вводятся с консоли.

**На 6-7 баллов:**

Все функции получают все необходимые данные (если они необходимы) через регистры a0 - a7, используют только регистры t (стек не используется), а также внутри функций не используются данные не переданные им (т.е. они атомарны). Что делает их соответствующими конвенции. Запись данных идет через буфер размером 512 байт в кучу. Ввод, вывод и обработка данных реализуется через отдельные подпрограммы.

**Автоматическое тестирование:**

Автоматическое тестирование проходит по всем файлам testx.txt заменяя x корректным числам и выводит результаты в соответствующий файл resx.txt

**На 8 баллов:**

Реализовано автотестирование. Добавлена опция для вывода результатов в консоль.

**На 9 баллов:**

Выполнены требования по использованию макросов - оберток (подробнее см. описание макро). Реализована дополнительная тестовая программа.

**На 10 баллов:**

Программа разбита на 3 различных ассемблерных файла (подробнее см. описание программы). Подпрограммы print\_string, print\_double, read\_int - унифицированные модули, которые вызываются повторно в обоих режимах работы программы.

Макро лежат в библиотеке макро в отдельном файле, подпрограммы лежат в отдельном ассемблерном файле. Используются дополнительные графические диалоговые окна для ввода.