**Problem 1:**

Sum the numbers entered from the keyboard until the input value is 0

**Program 2:**

|  |  |
| --- | --- |
| **Related knowledge** | Use the function **getchar()** –stdio.h**,** to input a character, the function **toupper(ch)** to convert a character to uppercase - **ctype.h**  ASCII code of the ENTER key: ‘\n’ |
| **Problem** | Write a C program that will:   * permit user inputting a string of characters. The input operation will terminate if the ENTER key is stroked. * print out the number of vowels, number of consonants, and number of others to the monitor. |
|  |  |

**Program 3:**

|  |  |
| --- | --- |
| **Objectives** | Practice implementing a program with simple menu. |
| **Related knowledge** | None |
| **Problem** | Write a C program that will execute repetitively using a simple menu as following:   1. **Process primes** 2. **Print min, max digit in an integer;** 3. **Quit**   **Select an operation:**   1. When user selects the option 1, the program will accept a positive integral number and print out a message about whether the input number is a prime or not. 2. When user selects the option 2, the program will accept a positive integral number and print out the minimum and maximum digit in this number. 3. The program will terminate when user selects the option 3. |
| **Analysis** | **Nouns:**  - positive integral number 🡪 **int n**  - A number represents a choice of user 🡪 **int choice;**  **Functions**:  **int prime( int n) 🡪 see above**  **void printMinMaxDigits( int n) 🡪 see above** |

**Problem 4:**

Make menu like that

Input name, age, Mobile number, birthday

############################################################

: Name: Son Go Ku

: age: 100

: Mobile number: 113

: Birthday(d/m/y): 19/6/1984

############################################################

Output: Son Go Ku 100 old Birthday is 19/6/1984 Mobile number: 113

**Problem 5:**

Develop a C-program that helps user managing an 1-D array of real numbers(maximum of 100 elements) , with initial number of elements is 0, using the following simple menu:

1- Add a value

2- Search a value

3- Print out the array

4- Print out values in a range

5- Print out the array in ascending order

Others- Quit

* When the option 1 is selected, user will enters a value then it is added to the array
* When the option 2 is selected, user will enters a value then number of it’s existences will be printed out.
* When the option 3 is selected, values in the array will be printed out.
* When the option 4 is chosen, user will enter 2 values, minVal and maxVal, the values in array which are between minVal and maxVal are printed out (minVal <=value<=maxVal)
* When the option 5 is chosen, values in array will be printed out in ascending order but **their position are preserved. ( sorting based their pointers only)**