

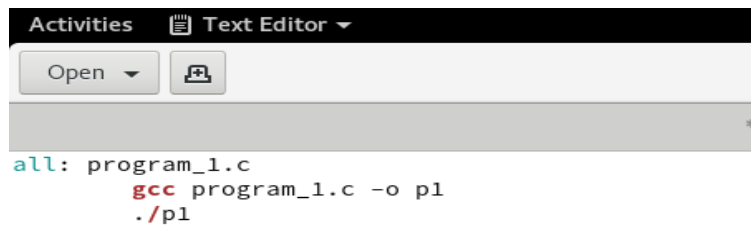
Andrew Baca
CS 471 Program 1
August 20, 2018

Program 1.c:

```
/******  
*  
* Andrew Baca  
* CS 471  
* Programming Assignment 1  
*  
* August 15th, 2018  
*  
* Procedure- This program will fill an array of 100 integers,with the ascii values of each letter of  
*           my name, and cast the array as a string to be printed out. This program is meant to test the  
*           byte sequence of C, with proper NULL termination. We are testing to see if the machine we are  
*           using is in a big endian ora little endian sequence.  
*  
* Input: An integer array with 100 values, filled with the ascii values in decimal, that would  
*        equivelate to my first and last name with proper spacing and indentation.  
*  
* Output: A string stored in the character pointer address with my first and last name with  
*         proper spacing and indentation,stored as little endian.  
*  
*****/  
#include <stdio.h>  
  
int main(void)  
{  
    int name[100];           //declare the array of 100 ingtegers  
    char *s;                 //declare a char pointer to store the name in  
  
    name[0] = (65 * 1) + (110 * 256) + (100 * 256 * 256) + (114 * 256 * 256 * 256); //Store name in ascii value as little endian  
    name[1] = (101 * 1) + (119 * 256) + (32 * 256 * 256) + (66 * 256 * 256 * 256);  
    name[2] = (97 * 1) + (99 * 256) + (97 * 256 * 256);  
    name[3] = 0;             //terminating value  
  
    s = (char*)name;         //cast the name array and store in the address of the char  
  
    printf("My name is %s\n", s); //print name  
  
}
```

Note : you store four letters per array space because we are using a 32 bit language, and each char is 8 bits

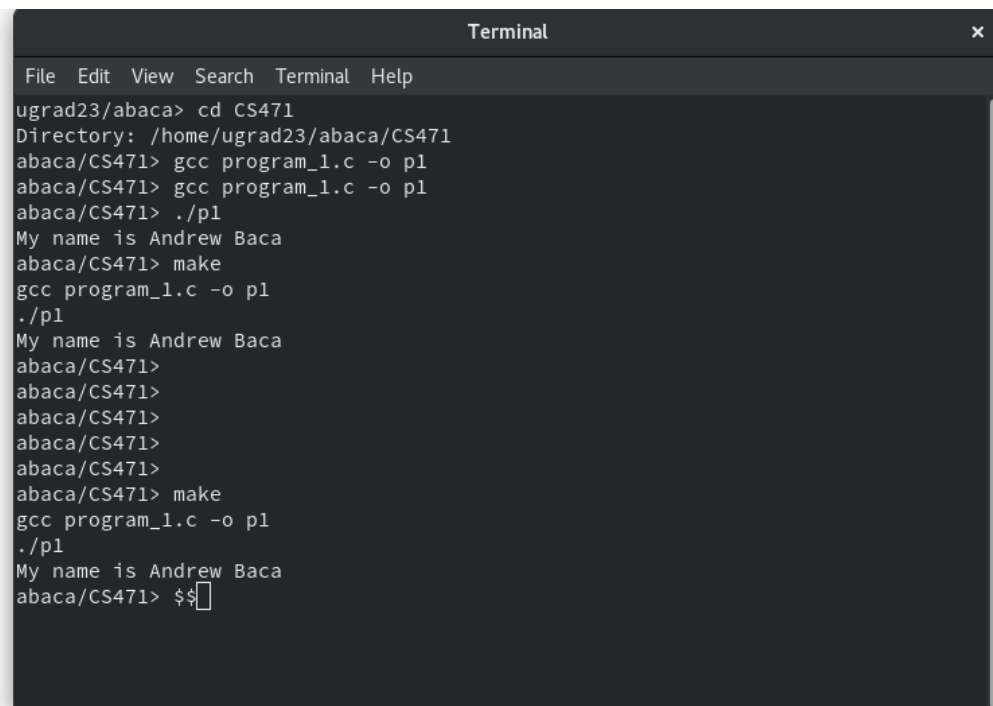
Makefile:



The screenshot shows a window titled "Text Editor" with a menu bar containing "Activities" and "Text Editor". Below the menu bar is a toolbar with an "Open" button and a file icon. The main text area contains the following Makefile content:

```
all: program_1.c
    gcc program_1.c -o p1
    ./p1
```

Output:



The screenshot shows a terminal window titled "Terminal" with a menu bar containing "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal output is as follows:

```
ugrad23/abaca> cd CS471
Directory: /home/ugrad23/abaca/CS471
abaca/CS471> gcc program_1.c -o p1
abaca/CS471> gcc program_1.c -o p1
abaca/CS471> ./p1
My name is Andrew Baca
abaca/CS471> make
gcc program_1.c -o p1
./p1
My name is Andrew Baca
abaca/CS471>
abaca/CS471>
abaca/CS471>
abaca/CS471>
abaca/CS471> make
gcc program_1.c -o p1
./p1
My name is Andrew Baca
abaca/CS471> $$
```