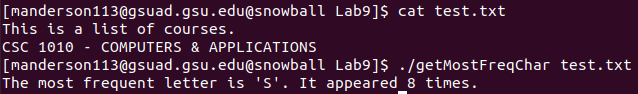
**Part 1**



#include <stdio.h>

int main(int argc, char\*\* argv) {

if (argc < 2)

{

printf("Please include valid file as an argument.\n");

return 1;

}

FILE \*inputFile;

inputFile = fopen(argv[1], "r");

if (inputFile == NULL)

{

printf("%s is not a valid file.\n", argv[0]);

return 1;

}

const int CAPS\_DIFF = ('a' - 'A');

int letters[26] = {0};

int character;

int most = 0;

int mostPosition = -1;

do

{

character = fgetc(inputFile);

character = (character > ‘Z’) ?  
 (character - CAPS\_DIFF) : character;

if ( character >= ‘A’ && character <= ‘Z’ )

{

int position = character – ‘A’;

letters[position] += 1;

int count = letters[position];

if (count > most)

{

most = count;

mostPosition = position;

}

}

} while (character != EOF);

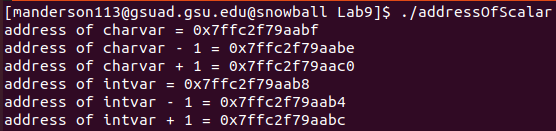
printf("The most frequent letter is '%c'. It appeared %d times.\n",

mostPosition + ‘A’, most);

return 0;

}

**Part 2**



#include <stdio.h>

#include <string.h>

int main() {

// initialize a char variable, print its address and the next address

char charvar = '\0';

printf("address of charvar = %p\n", (void \*) (&charvar));

printf("address of charvar - 1 = %p\n", (void \*) (&charvar - 1));

printf("address of charvar + 1 = %p\n", (void \*) (&charvar + 1));

// initialize an int variable, print its address and the next address

int intvar = 1;

printf("address of intvar = %p\n", (void \*) (&intvar));

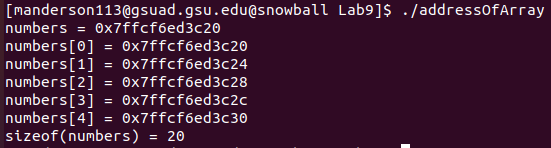
printf("address of intvar - 1 = %p\n", (void \*) (&intvar - 1));

printf("address of intvar + 1 = %p\n", (void \*) (&intvar + 1));

}

Each int takes up four bytes, so to reach the next integer it must increment by 4 instead of 1. In fact, it is incrementing by the sizeof of the first element in the array, which could be something other than 4 depending on the data type and implementation.

**Part 3**



#include <stdio.h>

#include <string.h>

int main() {

// initialze an array of ints

int numbers[5] = {1,2,3,4,5};

int i = 0;

// print the address of the array variable

printf("numbers = %p\n", numbers);

// print addresses of each array index

do {

printf("numbers[%u] = %p\n", i, (void \*) (&numbers[i]));

i++;

} while (i < 5);

// print the size of the array

printf("sizeof(numbers) = %lu\n", sizeof(numbers));

printf("sizeof(numbers) / sizeof(numbers[0]) = %d\n",

sizeof(numbers) / sizeof(numbers[0]));

}

The address of the array is always the same as the address of the first element of the array.

To find the length of the array, you can divide the size of the array (number of bytes in the array) by the size of the first element in the array (number of bytes in an element of the array).

**sizeof(numbers) / sizeof(numbers[0])**