Pointers

Dynamic Memory Allocation
malloc(), calloc() & free() functions

malloc()

- The name "malloc" stands for memory allocation.
- The malloc() function reserves a block of memory of the specified number of bytes.
- It returns a pointer of void which can be casted into pointers of any form.
- Syntax of malloc()

```
ptr = (castType*) malloc(size);
```

```
ptr = (float*) malloc(100 * sizeof(float));
```

The above statement allocates 400 bytes of memory. It's because the size of float is 4 bytes. And, the pointer ptr holds the address of the first byte in the allocated memory.

free()

- Dynamically allocated memory created with either calloc() or malloc() doesn't get freed on their own. You must explicitly use free() to release the space.

- Syntax of free()

  free(ptr);

- This statement frees the space allocated in the memory pointed by ptr.

Example 1: malloc() and free()

```
// Program to calculate the sum of n numbers
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    ptr = (int*) malloc(n * sizeof(int));

   if(ptr == NULL)
    {
        printf("Error! memory not allocated.");
        exit(0);
    }
    printf("Enter elements: ");
    for(i = 0; i < n; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }

    printf("Sum = %d", sum);

    // deallocating the memory
    free(ptr);

    return 0;
}
```

calloc()

- The name "calloc" stands for contiguous allocation.
- The malloc() function allocates memory and leaves the memory uninitialized. Whereas, the calloc() function allocates memory and initializes all bits to zero.
- Syntax of calloc()

  ptr = (castType*)calloc(n, size);
- Example:

  ptr = (float*) calloc(25, sizeof(float));
- The above statement allocates contiguous space in memory for 25 elements of type float.

calloc() and free()

```
// Program to calculate the sum of n numbers
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    ptr = (int*) calloc(n, sizeof(int));
    if(ptr == NULL)
    {
        printf("Error! memory not allocated.");
        exit(0);
    }
    printf("Enter elements: ");
    for(i = 0; i < n; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }

    printf("Sum = %d", sum);
    free(ptr);
    return 0;
}
```

Homework

What is wrong with the following program

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {

  int *a = malloc(sizeof(int));
   *a = 10;

  printf("%d\n", *a);

  a = calloc(3, sizeof(int));
  a[0] = 10;
  a[1] = 20;
  a[2] = 30;

  printf("%d %d %d\n", a[0], a[1], a[2]);

  return 0;
}
```