```
#include <unistd.h>
int main(){
write(1,” Hi! My name is pw\n”,20)
}

#include <unistd.h>

#define STDOUT_FILENO 1
#define STDOUT_FILENO 2
int main(){
	int count;
	int i;
	for(count = 0; count < 3; count++){
		for(i = 0; i < 3-count; i++){
			write(STDOUT_FILENO,"*",4);
		}
		write(STDOUT_FILENO," ",3);
	}
	return 42;
}

#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#define STDOUT_FILENO 1
#define STDOUT_FILENO 2
int main(){
	int count;
	mode_t mode = S_IRUSR | S_IWUSR;
	int fildes = open("output.txt",O_CREAT | O_TRUNC | O_RDWR,mode);
	write(fildes, "Great!\n",7);
	close(fildes);
	for(count = 0; count < 3; count++){
		write(STDOUT_FILENO,"Hello World\n",20);
		write(STDOUT_FILENO,".",1);
	}
	return 42;
}


#include <unistd.h>
#include <sys/types.h>
```

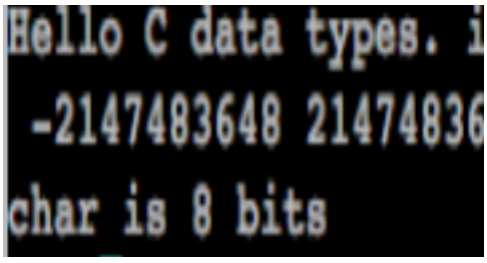
```
#include <sys/stat.h>
#include <fcntl.h>

#include <stdio.h>

//#define STDOUT_FILENO 1
//#define STDOUT_FILENO 2
int main(){
	int count;
	//int result = 65;
	//printf("My value is %d\n",result);
	mode_t mode = S_IRUSR | S_IWUSR;
	//close(1);
	int fildes = open(".",O_CREAT | O_TRUNC | O_RDWR,mode);
	if(fildes == -1){
		perror("open failed");
		exit(1);
	}
	printf("fildes is %d\n", fildes);
	write(fildes, "Great!\n",7);

	for(count = 5; count; count--){
		write(STDOUT_FILENO,"Hello\n",6);
		write(STDOUT_FILENO,".",1);
	}
	close(fildes);
	return 42;
}
```

Not all bytes are 8 bits?

- How many bits are there in a byte?8
- How many bytes is a `char`?1
- 

Tell me how many bytes the following are on your machine: `int, double, float, long, long long`

4,8,4,4,8

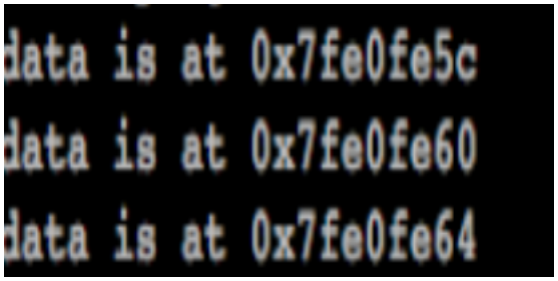
```
#include <stdio.h>
```

```
int main(){
    int data[8];
    printf("data is at %p \n",data);
    printf("data is at %p \n",data+1);
    printf("data is at %p \n",data+2);

    *(data) = 100;
    *(data+1) = 101;
    *(data+2) = 102;

    data[2] = 102;
    2[data] = 102;//crazy

    return 0;
}
```




```
#include <stdio.h>
#include <string.h>
int main(){
    printf(" Size of hello is %d\n",sizeof("hello"));

    char *ptr = "hello";
    *ptr = 'J';
    int len = strlen(ptr);
    int count = 0;
    while( *ptr ){
        printf("%c", *ptr);
        ptr ++;
        count++;
    }
    printf("Characters we found: %d\n",count);

    printf("Strlen returned %d\n",len);
    return 0;
}
```

Why does this segfault?

- `char *ptr = "hello";`
- `*ptr = 'J';`
- try to change first character to be J, different parts of my processes are valid for reading and writing, the program code exact assemble the instruction for cpu can only be read, and my constant such as hello can only be read and the hardware if it fixed enough to know which part of memory are read and invalid.

What does `sizeof("Hello\0World")` return?12

What does `strlen("Hello\0World")` return?5

Give an example of X such that `sizeof(X)` is 3: sizeof("ab")

Give an example of Y such at `sizeof(Y)` might be 4 or 8 depending on the machine. **pointer**

char *ptr = "hello";

printf(" Size of hello is %d\n",sizeof(ptr));

Name me two ways to find the length of `argv`

```
int main(int argc, char* argv[]) {
	printf("argv[0]=%s\n",argv[0]);
	printf("argv[argc]=%p\n",argv[argc]);
	printf("argc = %d\n",argc);
	int count = 1;
	int sum = 0;
	for(;count < argc;count--){
		//printf("%d :%s",count, argv[count]);
		sum += atoi(arhv[count]);
	}
	return 0;
}
```

What is `argv[0]`

The full path name of the program run

```
/*Write your C code here*/
#include <stdio.h>

extern char** environ;
int main() {
	char * secret = getenv("VERYSECRET");
	printf("The secret is %d, %s\n",secret);
	if(secret == NULL){
		exit(0);
```

```
    }
    char ** ptr = environ;
    while(*ptr){
        printf("%s\n",*ptr);
        ptr++;
    }
    return 0;
}
```

Environment variables are stored in a special array that can be read by your `main` function.USER=user

```
/*Write your C code here*/
#include <stdio.h>
char* change(char*);

int main() {
    //char * ptr = "Hello";
    char array[] = "Hello World";
    // printf("%d %d\n",sizeof(ptr),sizeof(array));
    char* result = change(array);
    printf("%s\n",result);
    return 0;
}

char* change(char* p){
    while(*p){
        //if(*p == 'l') *p = '*';
        if(*p == 'o') return p;
        p++;
    }
    return p;
}
```

What is the results of `sizeof(ptr)` and `sizeof(array)`? Explain why.

4,6

the size of pointer is num of bits exctraly hold character pointer

the size of array is total size require to hold the array

```
/*Write your C code here*/
#include <stdio.h>
#include <stdlib.h>

char* f1( ){
    char array[] = "f1f1";
    printf("f1 : %p\n",array);
    //if(level) f1(level -1);
    return array;
}
```

```
/*void f2(){
char array[] = "f2f2";
printf("f2 : %p\n",array);
}
*/
/*
void eg(){
char blah[1024];
f1();
}*/
int main() {
	char * ptr = f1();
	//eg();
	return 0;
}
```

What datastucture is managing the lifetime of automatic variables?
Stack

```
/*Write your C code here*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
//static char array[256];

char* currenttime(){
	char* result = malloc(128);
	//calloc,re
	if(!result) return result;
	//static char array[256];
	strcpy(result, "2:51 PM");
	return result;
}

int main() {
	char* ptr = currenttime();
	printf("%s",currenttime());
	free(ptr);
	return 0;
}
```

- If I want to use data after the lifetime of the function it was created in, then where should I put it and how do I put it there?static at the front of the data declare

Fill in the blank. In a good C program: "For every malloc there is a _free__".

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

```
#include <stdlib.h>
#include <string.h>
#include <time.h>
//static char array[256];

char* currenttime(){
	char* result = malloc(128);
	//calloc,re
	if(!result) return result;
	time_t secondsSince1970 = time(NULL);
	char* timeASCII = ctime( & secondsSince1970);
	//static char array[256];
	strcpy(result, timeASCII);
	return result;
}

int main() {
	char* ptr = currenttime();
	free(ptr);
	ptr = NULL;
	printf("%s",ptr);
	//free(ptr);
	//free(ptr);
	return 0;
}
```

Name one reason `malloc` can fail

- **Input validation**. For example, you've asked for many gigabytes of memory in a single allocation. The exact limit (if any) differs by malloc implementation; POSIX says nothing about a maximum value, short of its type being a size_t.
- **Allocation failures**. With a data segment-based malloc, this means **brk** failed. This occurs when the new data segment is invalid, the system is low on memory, or the process has exceeded its maximum data segment size (as specified by RLIMIT_DATA). With a mapping-based malloc, this means **mmap** failed. This occurs when the process is out of virtual memory (specified by RLIMIT_AS), has exceeded its limit on mappings, or has requested too large a mapping. Note most modern Unix systems use a combination of brk and mmap to implement malloc, choosing between the two based on the size of the requested allocation.

You asked for a zero-sized allocation. Per POSIX, malloc *may* return **NULL** if you call malloc with a parameter of zero.

Name some differences between `time()` and `ctime()`

Get current time (function)

Convert time_t value to string (function)

ctime is the inode or file change time. The ctime gets updated when the file attributes are changed, like changing the owner, changing the permission or

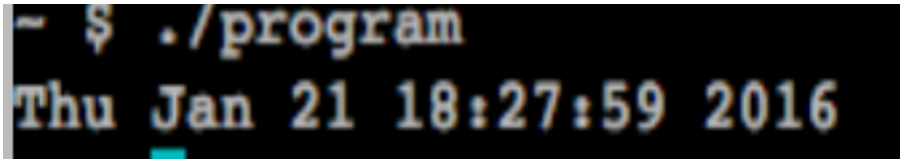
moving the file to an other filesystem but will also be updated when you modify a file.

◦ What is wrong with this code snippet?

```
free(ptr);
free(ptr);free a second time
```

◦ What is wrong with this code snippet?

```
free(ptr);
printf("%s\n", ptr);free a invalid memory
after free set it to NULL,and one malloc one free
```




```
/*Write your C code here*/
#include <stdio.h>


struct Link{
	int value;
	struct Link* next;
}
//typedef int number;
//number x;

typedef struct Link link_t;

//Link_t one;
//int value;
int main(){
	//	one.value = 42;
	//		one.next = &one;
	link_t * ptr1 = (link_t*)malloc(sizeof(link_t));
	link_t * ptr2 = (link_t*)malloc(sizeof(link_t));
	ptr1->value = 42;
	ptr1->value = 42;
	ptr1->next = ptr2;
	ptr1->next = NULL;
	return 0;
}

/*Write your C code here*/
#include <stdio.h>
```

```
struct Person{
	string name;
	int age;
	struct Link * friends;
	//struct Link* next;
}
//typedef int number;
//number x;

typedef struct Link link_t;

//Link_t one;
//int value;
int main(){
	//    one.value = 42;
	//        one.next = &one;
	link_t * ptr1 = (link_t*)malloc(128);
	link_t * ptr2 = (link_t*)malloc(256);
	ptr1->name = "Agent Smith";
	ptr2->name = "Sonny Moore";
	ptr1->friends = ptr2;
	ptr2->friends = ptr1;
	free(ptr1);
	free(ptr2);
	return 0;
}

/*Write your C code here*/
#include <stdio.h>


struct Person{
	char* key;
	char* value;
	struct Link * next;

}

typedef struct Link link_t;

link_t * link_create(char akey,char* avalue){
	printf("Creating link %s -> %s",akey, avalue);
```

```
    link_t *result = (link_t*)malloc(sizeof(link_t));
    result -> key = strdup(akey);
    result -> value = strduo(avalue);
}

void link_destroy(link_t*p){
    free(p->ley);
    free(p->value);
    memset(p, 0, sizeof(link_t));
    free(p);
}

link_t root;
int main(){
    return 0;
}
```

What functions can be used for getting characters for `stdin` and writing them to `stdout`?

File Streams

`main` function

Reads characters from the *standard input* (stdin) and stores them as a C string into *str* until a newline character or the *end-of-file* is reached.

The newline character, if found, is not copied into *str*.

String a,b,c;

`sscanf` ("Hello", 5, "World", “%S %S %S”,a,b,c)

- What does one need to define before using `getline()`?
  - `#include <iostream>`
  - `#include <string>`
-

Write a C program to print out the content of a file line by line using `getline()`

```
#include <iostream>
#include <string>

int main ()
{
  std::string name;

  std::cout << "Please, enter your full name: ";
  std::getline (std::cin,name);
  std::cout << "Hello, " << name << "!\n";
```

```
return 0;}
```

- What compiler flag is used to generate a debug build?
- You modify the makefile to generate debug builds and type `make` again. Explain why this is insufficient to generate a new build.
- Are tabs or spaces used in Makefiles?yes
- What are the differences between heap and stack memory? Stack is used for static memory allocation and Heap for dynamic memory allocation, both stored in the computer's RAM .
- Are there other kinds of memory in a process?

Optional (Just for fun)

- Convert your a song lyrics into System Programming and C code covered in this wiki book and share on Piazza.
- Find, in your opinion the best and worst C code on the web and post the link to Piazza

Write a short C program with a deliberate subtle C bug and post it on Piazza to see if others can spot your bug