Exercise 2

Kendrick Kwong

randomNum.h:

```
randomNum.h
#ifndef RANDOMNUM_H
#define RANDOMNUM_H

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

int * randomNum(int max, int min, int size);

#endif
~
~
~
~
~
```

randomNum.c

```
randomNum.c

#include <time.h>
#include <stdlib.h>
#include <stdio.h>
#include "randomNum.h"
#include <malloc.h>

int * randomNum(int max, int min, int size){
    srand(time(NULL));
    int *r = malloc(sizeof(size));

    for(int i=0 ; i<size ; ++i){

        r[i] = (rand() % (max - min)) + min;

    }
    free(r);
    return r;
  }

~
~
```

Main.c:

~/lab 2/r

main.c

```c
#include <setjmp.h>
#include <signal.h>

static jmp_buf jbuf;
static void catch_segv()
{
    longjmp(jbuf, 1);
}





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


    signal(SIGSEGV, catch_segv);
    if(setjmp(jbuf) == 0){
        int max = atoi(argv[1]);
        int min = atoi(argv[2]);
        int size = atoi(argv[3]);
        int *p;
    if(min >= max || size == 0 || size >(max-min)){
        printf("Oops! You entered wrong format!\n");
    }else{
            printf("Random numbers are:\n");
            p = randomNum(max,min,size);
            for(int i=0 ; i < size; ++i){
                printf(" %d ",p[i]);
            }
        }
    }else{
        printf("Oops! You entered wrong format!\n");
    }
}
```

Function test:

```
kendrick807@kendrick807-virtual-machine  ~/lab 2
./main 3 2 4
Oops! You entered wrong format!
kendrick807@kendrick807-virtual-machine  ~/lab 2
./main 3 2 1
Random numbers are:
 2 %
rtual-machine  ~/lab 2
./main 3 2 0
Oops! You entered wrong format!
kendrick807@kendrick807-virtual-machine  ~/lab 2
./main 10 2 1
Random numbers are:
 9 %
rtual-machine  ~/lab 2
./main 10 2 5
Random numbers are:
 2  6  8  4  4 %
rtual-machine  ~/lab 2
./main 2 10 5
Oops! You entered wrong format!
kendrick807@kendrick807-virtual-machine  ~/lab 2
./main
Oops! You entered wrong format!
kendrick807@kendrick807-virtual-machine  ~/lab 2
./main 10 2
Oops! You entered wrong format!
kendrick807@kendrick807-virtual-machine  ~/lab 2
```

Preprocessor:

To invoke GCC and only carry out the preprocessor command:

```
kendrick807@kendrick807-virtual-machine  ~/lab 2
gcc -E randomNum.c main.c
# 0 "randomNum.c"
# 0 "<built-in>"
```

gcc -E in.c -o in.i
copy the output of the preprocessor to another .c file

The difference between my .c files and what they look like after the preprocessor has done its thing is that it will generation an object file (i.e_randomNum.o , main.o)

Compilation: