

Exercise 2

Kendrick Kwong

Script:

randomNum.h:

```
randomNum.h
1 #ifndef RANDOMNUM_H
2 #define RANDOMNUM_H
3
4 #include <time.h>
5 #include <stdlib.h>
6 #include <stdio.h>
7
8 int randomNum(int max, int min);
9
10 #endif
```

randomNum.c

```
randomNum.c
1
2 #include <time.h>
3 #include <stdlib.h>
4 #include <stdio.h>
5 #include "randomNum.h"
6
7 int randomNum(int max, int min){
8     int r = 0;
9     r = (rand() % (max - min)) + min;
10    return r;
11 }
12
```

Main.c:

```
1
2 #include "randomNum.h"
3 #define INT_SIZE 3
4 #include <setjmp.h>
5 #include <signal.h>
6
7 static jmp_buf jbuf;
8 static void catch_segv()
9 {
10     longjmp(jbuf, 1);
11 }
12
13 void scanNum(){
14     int max = 0;
15     int min = 0;
16     int size = 0;
17     printf("Oops! You entered wrong format!\n");
18     printf("Enter numbers by format max min range:\n");
19     if(scanf("%d %d %d",&max,&min,&size) != 3 || min >= max || size == 0 || size > (max-min+1)){
20         printf("Oops! You entered wrong format AGAIN!\n");
21     }else{
22         int p[size];
23         printf("Random numbers are:\n");
24         for(int i=0;i<size;i++){
25             p[i] = randomNum(max,min);
26         }
27         for(int i=0 ; i < size; ++i){
28             printf(" %d ",p[i]);
29         }
30     }
31 }
32
33 }
```

```
34
35 int main(int argc, char** argv){
36     srand(time(NULL));
37
38     signal(SIGSEGV, catch_segv);
39     if(setjmp(jbuf) == 0){
40
41         int max = atoi(argv[1]);
42         int min = atoi(argv[2]);
43         int size = atoi(argv[3]);
44         int p[size];
45
46         printf(" %d ",size);
47         if(min >= max || size == 0 || size > (max-min+1)){
48             scanNum();
49         }else{
50             printf("Random numbers are:\n");
51             for(int i=0;i<size;i++){
52                 p[i] = randomNum(max,min);
53             }
54             for(int i=0 ; i < size; ++i){
55                 printf(" %d ",p[i]);
56             }
57         }
58     }else{
59         scanNum();
60     }
61 }
62
63 }
```

Preprocessor:

To invoke GCC and only carry out the preprocessor command:

```
kendrick807@kendrick807-virtual-machine > 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 generate an object file (i.e. randomNum.o , main.o)

Compilation:

Command to compile:

```
kendrick807@kendrick807-virtual-machine > gcc -Wall -Wextra -Wfatal-errors -Wpedantic main.c randomNum.c
kendrick807@kendrick807-virtual-machine >
```

-Wall

Enables all compiler's warning messages. This option should always be used, in order to generate better code.

-Wextra

Warn if a comparison is always true or always false due to the limited range of the data type, but do not warn for constant expressions. For example, warn if an unsigned variable is compared against zero with '<' or '>='. This warning is also enabled by

-Wfatal-errors

This option causes the compiler to abort compilation on the first error occurred rather than trying to keep going and printing further error messages.

-Wpedantic

GCC compilers always try to compile your program if this is at all possible. However, in some cases, the C and C++ standards specify that certain extensions are forbidden. Conforming compilers such as GCC or g++ must issue a diagnostic when these extensions are encountered.

the GCC compiler's -pedantic option causes GCC to issue warnings in such cases.

When the command -Ofast is used instead of -O, the compile time increased but the execution time decreased.

Linking:

Command:

```
kendrick807@kendrick807-virtual-machine > gcc -o main randomNum.c main.c
```

Execution and debugging:

Command:

```
➤ ./mains
Oops! You entered wrong format!
Enter numbers by format max min range:
12 2 0
Oops! You entered wrong format AGAIN!
kendrick807@kendrick807-virtual-machine > ./lab2/embedded-linux/exercise_3/3a & & main
➤ ./mains 100 20 120
120 Oops! You entered wrong format!
Enter numbers by format max min range:
1202 24 23
666 371 496 480 735 820 1108 929 28 395 31 1044 250 897 645 988 193 413 943 1153 625 473 1129
kendrick807@kendrick807-virtual-machine > ./lab2/embedded-linux/exercise_3/3a & & main
➤ ./mains 100 20 12
12 Random numbers are:
89 42 27 22 66 82 92 75 90 49 29 91
```