CS 2263 - FR01A

Lab 6

By Ngoc Phuong Anh Nguyen - 3712361

November 2021

Question 1:

Create a program based on the genPointsBin.c program from lecture that creates a binary file of random integer values and outputs them to a file specified on the command line. The first value in the file should be the number of values in the file (yes, as a binary value), followed by binary integer values (one after another – no spaces, commas, line feeds – just the binary values).

```
$ genIntBin 10000 16ex1.bin
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

char* decimalToBinary(int n)
{
  int i;
  static char binary[16];
  for(i = 15; i >= 0; i--)
  {
    if(n%2 == 0)
    {
      binary[i] = '0';
    }
    else
    {
      binary[i] = '1';
    }
    n/=2;
  }
  return binary;
```

Figure 1: Source Code Of Question 1

```c
}

int main(int argc, char* argv[])
{
  srand(time(0));
  int i,j;
  int quantity = atoi(argv[1]);
  FILE* fileWrite = fopen(argv[2], "wb");
  char* out = decimalToBinary(quantity);
  fwrite(&out,sizeof(char), strlen(decimalToBinary(quantity)),fileWrite);
  for(j = 0; j < quantity; j++)
  {

    int value = rand()%quantity;
    char* output = decimalToBinary(value);
    fwrite(&output,sizeof(char), strlen(output),fileWrite);
  }
  fclose(fileWrite);

  return EXIT_SUCCESS;
}
```

Figure 2: Source Code Of Question 1

```
[anguyen5@gc112m30 Lab 6]$ make
gcc genIntBin.c -o genIntBin
[anguyen5@gc112m30 Lab 6]$ ./genIntBin 1000 16ex1.bin
[anguyen5@gc112m30 Lab 6]$
```

Figure 3: The make command output for a successful compile and the Output Result Of Question 1

Question 2 (Incomplete):

Create a program that reads the binary file from a name specified on the command line into a heap allocated array in memory and sorts it in place. Use whatever sorting algorithm you wish, so long as you write it yourself as a function and name it in your comments. Your function will take in a pointer to the array and it's filled length. Incorporate the technique from lecture of calculating elapsed time and measure the time it takes to sort the integer array. The program should write the values out to a binary file l6ex2.bin once sorted.

```
$ sortInMemoryIntBin l6ex1.bin l6ex2.bin
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>
char* decimalToBinary(int n)
{
  int i;
  static char binary[16];
  for(i = 15; i >= 0; i--)
  {
    if(n%2 == 0)
    {
      binary[i] = '0';
    }
    else
    {
      binary[i] = '1';
    }
    n/=2;
  }
  return binary;
}

int binaryToDecimal(char* n)
```

Figure 4: Source Code Of Question 2

```
{
  int i, j=15;
  int decimal = 0;
  for(i = 0; i < 16; i++)
  {

    if(n[i] == '1')
    {
      decimal += pow(2,j);
    }
    j--;
  }
  return decimal;
}


int main(int argc, char* argv[])
{
  char* bin = "0000000000101110";
  printf("%d",binaryToDecimal(bin));

  return EXIT_SUCCESS;
}
```

Figure 5: Source Code Of Question 2

```
[anguyen5@gc112m30 Lab 6]$ make
gcc genIntBin.c -o genIntBin
gcc -lm sortInMemoryIntBin.c -o sortInMemoryIntBin
[anguyen5@gc112m30 Lab 6]$ ./sortInMemoryIntBin
46[anguyen5@gc112m30 Lab 6]$
```

Figure 6: The make command output for a successful compile and the Output Result Of Question 2

```
GCC = gcc
TARGETS = genIntBin sortInMemoryIntBin
CFLAGS = -lm

all: $(TARGETS)

genIntBin:
	$(GCC) $@.c -o $@

sortInMemoryIntBin:
	$(GCC) $(CFLAGS) $@.c -o $@

%.o: %.c
	$(GCC) -c $*.c

clean:
	rm -f $(TARGETS) *.o
```

Figure 7: Makefile for the whole Lab