**Exercise 1**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include "Point.c"

int main(int argc, char const \*\*argv)

{

    if(argc != 2)

    {

        printf("%s : %s\n", argv[0], "<number of points>");

        return EXIT\_FAILURE;

    }

    int var = atoi(argv[1]);

    Point\* pPt = mallocPoint();

    for(int i = 0; i < var; i++)

    {

        pPt->id = i;

        pPt->x = rand()/(float)10000;

        pPt->y = rand()/(float)10000;

        putw(var, stdout);

        fwritePoint(stdout, pPt);

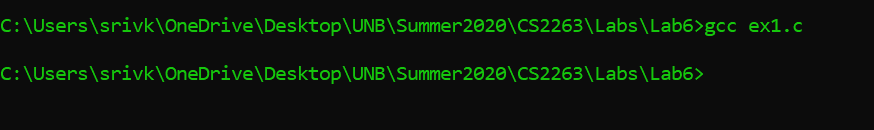
    }

    freePoint(pPt);

    fprintf(stderr, "Binary file made successfully. \n");

    return EXIT\_SUCCESS;

}



**Exercise 2**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <sys/time.h>

void sortNum(int arr[], int a, int b)

{

    int num1;

    int num2;

    int num3;

    int num4;

    if(a < b)

    {

        num1 = a;

        num2 = a;

        num3 = b;

        while(num2 < num3)

        {

            while(arr[num2] <= arr[num1] && num2 <= b)

            {

                num2++;

            }

            while(arr[num3] > arr[num1] && num3 >= 1)

            {

                num3--;

            }

            if(num2 < num3)

            {

                num4 = arr[num2];

                arr[num2] = arr[num3];

                arr[num3] = num4;

            }

        }

        num4 = arr[num3];

        arr[num3] = arr[num1];

        arr[num1] = num4;

        sortNum(arr, 1, num3-1);

        sortNum(arr, num3+1, b);

    }

}

int main(int argc, char\*\* argv)

{

    struct timeval start;

    struct timeval end;

    float done;

    gettimeofday(&start, NULL);

    int array[10];

    gettimeofday(&end, NULL);

    FILE \*fptr1 = fopen(argv[1], "rb");

    FILE \*fptr2 = fopen(argv[2], "wb");

    fread(array, sizeof(int)\*10, 1, fptr1);

    sortNum(array, 0, 9);

    printf("After sorting: \n");

    for(int i = 0; i < 10; i++)

    {

        printf("%d\n", array[i]);

        printf("\n");

    }

    fread(array, sizeof(int)\*10, 1, fptr2);

    done = (end.tv\_sec - start.tv\_sec) + 1e-6\*(end.tv\_usec - start.tv\_usec);

    printf("Time taken: %f\n", done);

    return 0;

}

Screenshots:

A picture containing indoor, sitting, holding, monitor

Description automatically generatedA close up of a logo

Description automatically generated

**Exercise 4**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main(int argc, char const \*\*argv)

{

    unsigned char ch[250];

    unsigned int var;

    int num;

    FILE \*fptr = fopen(argv[1], "rb");

    FILE \*fptr2 = fopen(argv[2], "w");

    fseek(fptr, 0, SEEK\_END);

    num = ftell(fptr);

    fseek(fptr, 0, SEEK\_SET);

    while(fread(&num, sizeof(char), 1, fptr) == 1)

    {

        fread(&ch, sizeof(char), 1, fptr);

        fread(&var, sizeof(int), 1, fptr);

        fprintf(stderr, "%s %d\n", ch, var);

    }

    fclose(fptr);

}

A picture containing object, indoor, clock, sitting

Description automatically generated