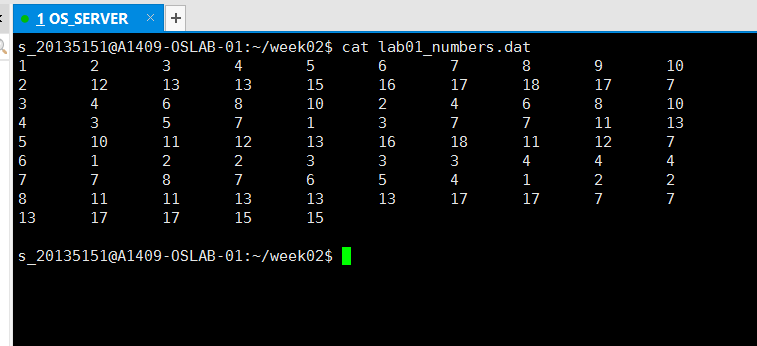
운영체제(2주차실습)

컴퓨터공학과 20135151 이갑성

1번



2번

코드

#include <stdio.h>

#include <stdlib.h>

#define MAX\_ELMNTS 100

#define ANLYS\_RNG 20

int getData(int numbers[], int size, int range); //데이터를 받는 함수원형

void printData(int numbers[], int size, int lineSize); //데이터를 프린트하는 함수원형

void makeFrequency(int numbers[], int size, int frequency[], int range); //얼마나 자주 있는지 기록하는 함수원형

void makeHistogram(int frequency[], int range); //기록을 만드는 함수원형

int main(void)

{

int size;

int nums[MAX\_ELMNTS]; //정수형 배열 num[100]

int frequency[ANLYS\_RNG]; //정수형 배열 frequency[20]

size = getData(nums, MAX\_ELMNTS, ANLYS\_RNG); //getData함수 리턴값은 배열에 저장된 인덱스값을 나타내는데 그 값을 size에 저장

printData(nums, size, 10); //배열에 가지고 있는 데이터 출력하는 함수

makeFrequency(nums, size, frequency, ANLYS\_RNG); //nums배열에 있는 값들의 빈도수를 frequency배열에 저장하는 함수

makeHistogram(frequency, ANLYS\_RNG); ////frequency배열에 있는 값(빈도수)를 출력하고 그 빈도수만큼 \*을 찍는 함수

return 0;

}

int getData(int data[], int size, int range)

{

int dataIn;

int loader = 0;

FILE \*fpData; //파일 포인터

range--; //range == 19

if (!(fpData = fopen("./lab01\_numbers.dat", "r"))) { //./lab01\_numbers.dat의 파일을 읽기 모드(r) 로 열기

printf("Error opening files \a\a\n"); // 조건문 fpData로 반환값이 false면 출력

exit(100); //함수 종료

}

while (loader < size && fscanf(fpData, "%d", &dataIn) != EOF) { //loader이 size(19)보다 작을때까지 그리고 dataln을 입력받는게 EndOfFile이 아닐때까지 라는 조건하에 반복문을 돌린다.

if (dataIn >= 0 && dataIn <= range) { //만약 dataln이 0이상이거나 19이하면

data[loader++] = dataIn; //data배열 loader인덱스 에 값을 넣어준다. 그리고 loader인덱스는 증가한다.

}

else {

printf("\nData point %d invalid. Ignored. \n", dataIn); //0미만이거나 19초과되는 수가 들어오면 문장출력

}

}

if (loader == size) { //인덱스를 나타내는 loader가 size(19) 값과 같아질때

printf("\nToo much data. Process what read.\n"); //해당 문장 출력

}

return loader; //인덱스를 나타내는 loader을 리턴해준다.

}

void printData(int data[], int size, int lineSize)

{

int numPrinted = 0; //줄 변경을 위한 변수

printf("\n\n");

for (int i = 0; i < size; i++) { //정수형 변수 i를 0으로 초기화하여 i를 배열의 size미만까지 반복문 실행

numPrinted++; //numPrinted를 증가

printf("%2d ", data[i]); //배열에 저장되있는 값을 출력

if (numPrinted >= lineSize) { // numPrinted가 lineSize(10) 보다 크거나 같아질때

printf("\n"); // 줄변경 작업

numPrinted = 0; //numPrinted는 초기화

}

}

printf("\n\n");

return;

}

void makeFrequency(int nums[], int last, int frequency[], int range)

{

for (int f = 0; f < range; f++) //정수형 변수 f를 0으로 초기화하여 f가 range(20)미만까지 반복문 실행

frequency[f] = 0; //frequency배열에 인덱스 0부터 19까지 값 0을 저장

for (int i = 0; i < last; i++) //정수형 변수 i에 nums배열의 사이즈 크기인 last값의 미만까지 반복문 실행

frequency[nums[i]]++; //nums배열에 i의 값이 들어가면 num[i]값을 frequency배열의 인덱스에 넣어서 그 값을 증가시켜준다.

//nums배열에 있는 값을 인덱스로 바꿔서 frquency배열에는 빈도수가 저장이 된다.

return;

}

void makeHistogram(int freq[], int range)

{

for (int i = 0; i < range; i++) {//정수형 변수 i를 0으로 초기화하여 i가 range(20)미만까지 반복문 실행

printf("%2d %2d ", i, freq[i]); //i와 freq배열에 있는 i번째 값을 출력한다.

for (int j = 1; j <= freq[i]; j++) //정수형 변수 j를 1로 초기화하여 j가 freq배열의 i번째 있는 값보다 작거나같을때까지

printf("\*"); //\*을 출력해준다.

printf("\n"); //줄바꿈

}

return;

}

실행결과

