운영체제(4주차실습)

컴퓨터공학과 20135151 이갑성

|  |
| --- |
| 코드(주석)  <main.c>  #include <stdio.h>  #include <stdlib.h>  #include <limits.h>  #include "main.h"  int main(void)  {  int \*\*table;  table = build\_table();  //행과 열을 메모리 할당한 뒤에 그 주소를 가르키는 주소값을 리턴하여 table에 저장  fill\_table(table);  //주소를 가르키는 table을 매개변수로 넣어 fill\_table함수 실행 => table의 각 열의 메모리에 값을 저장하는 함수  process\_table(table);  //주소를 가르키는 table을 매개변수로 넣어 process\_table함수 실행 => table에 각 행의 열의 메모리 값 중에서 최대 최소 평균값을 나타내는 함수  return 0;  }  ~  <tables.c>  #include <stdio.h>  #include <stdlib.h>  #include "tables.h"  //행을 입력받은 후 행에 대해서 입력받은 +1 만큼 동적 메모리 할당한 후 그 주소값을 table에 넣어준다.  //반복문을 통해 각 행마다 입력받은 열의 값에 +1한 만큼 동적 메모리 할당을 해주고, 각 행의 0번째 열의 메모리에는 열의 메모리 사이즈 값을 넣어준다.  //마지막 행의 메모리에는 NULL값을 넣어 마지막이라는 것을 나타내준다.  //그리고 메모리를 만들어진 것에 주소를 리턴해준다.  int\*\* build\_table(void)  {  int row\_num, col\_num;  int \*\*table;  int row;  printf("\nEnter the number of rows in the table: ");  scanf("%d", &row\_num);  table = (int \*\*)calloc(row\_num+1, sizeof(int \*));  for(row = 0; row < row\_num; row++){  printf("Enter number of integers in row %d: ", row+1);  scanf("%d", &col\_num);  table[row] = (int \*)calloc(col\_num+1, sizeof(int));  table[row][0] = col\_num;  }  table[row] = NULL;  return table;  }  void fill\_table(int \*\*table)  {  int row = 0;  printf("\n ===========================");  printf("\n Now we fill the table. \n");  printf("\n For each row enter the data");  printf("\n and press return: ");  printf("\n ===========================\n");  //table의 행의 메모리가 NULL을 가지고 있는 메모리를 만날때까지 반복문을 수행해주는데 반복문의 수행과정은  //table의 첫번째 행 메모리가 가르키는 열 메모리의 1번째 메모리부터 입력받은 값을 채워준다.  //그리고 row++을 통해 다음 행을 가르킬 수 있게한다. 두번째 행 메모리도 위와 동일한 수행과정을 통해 열 메모리 값을 넣어준다.  while(table[row] != NULL)  {  printf("\n row %d(%d integers) ====> ", row+1, table[row][0]);  for(int column = 1; column <= \*table[row]; column++){  scanf("%d", table[row]+column);  }  row++;  }  return;  }  void process\_table(int \*\*table)  {  int row = 0;  int row\_min, row\_max;  float row\_ave;  //행이 마지막 메모리에 있는 NULL을 만나기 전까지 반복문 수행한다. 수행과정은  //row\_min에는 각 행의 열의 메모리값들 중에서 가장 작은 열의 값을 저장하고  //row\_max에는 각 행의 열의 메모리값들 중에서 가장 큰 열의 값을 저장하고  //row\_ave에는 각 행의 열의 메모리값들의 평균값을 저장한다.  //그 저장된 값을 출력해주고 row를 증가시켜줌으로써 다음 행으로 옮겨 위와 동일한 수행과정을 거친다.  while(table[row] != NULL)  {  row\_min = row\_minimum(table[row]);  row\_max = row\_maximum(table[row]);  row\_ave = row\_average(table[row]);  printf("\nThe statistics for row %d ", row+1);  printf("\nThe minimum: %5d", row\_min);  printf("\nThe maximum: %5d", row\_max);  printf("\nThe average: %8.2f ", row\_ave);  row++;  }  printf("\n");  return;  }  <rows.c>  #include <limits.h>    #include "rows.h"  //row\_min변수에 정수형 최대값을 넣어주고 for문 반복문을 통하여 각 행의 열 1번째 값과 row\_min을 smaller함수에 매개변수로 넣어주고 작은 값을 리턴받아 row\_min에 넣어준다.  //다음 2번째 열의 값과 row\_min을 비교해주는 동일한 작업을 해주고 반복문을 수행 과정이 끝나면 최종적으로 row\_min에는 행의 열값들 중에 가장 작은 값이 들어간다.  //그 가장 작은 값이 저장된 row\_min을 리턴해주고 함수의 작업은 끝난다.  int row\_minimum(int \*row\_ptr)  {  int row\_min = INT\_MAX;  for(int column = 1; column <= \*row\_ptr; column++){  row\_min = smaller(row\_min, \*(row\_ptr+column));  }  return row\_min;  }  //row\_max변수에 정수형 최소값을 넣어주고 for문을 통해 각 행의 열 1번째 값과 row\_max을 larger함수에 매개변수로 넣어주고 큰 값을 리턴받아 row\_max에 넣어준다.  //그 다음 2번째 열의 값과 row\_max를 비교해주는 동일한 작업을 해주고 반복문 수행 과정이 끝나면 최종적으로 row\_max에는 행의 열값들 중에서 가장 큰 값이 들어간다.  //그 가장 큰 값이 저장된 row\_max을 리턴하고 함수의 작업은 끝난다.  int row\_maximum(int \*row\_ptr)  {  int row\_max = INT\_MIN;  for(int column = 1; column <= \*row\_ptr; column++){  row\_max = larger(row\_max, \*(row\_ptr+column));  }  return row\_max;  }  //평균값을 나타내는 함수라 변수를 float으로 설정해주고 행의 열의 값들을 모두 더해주고 total에 넣어준다.  //row\_ave에 total과 각 행의 열의 메모리 길이를 알 수 있는 \*row\_ptr을 나누어준다.  //그리고 평균값이 저장되어있는 row\_ave를 리턴해준다.  int row\_average(int \*row\_ptr)  {  float total = 0;  float row\_ave;  for(int column = 1; column <= \*row\_ptr; column++){  total += (float)\*(row\_ptr+column);  }  row\_ave = total / \*row\_ptr;  return row\_ave;  }  <ops.c>  #include <stdio.h>    //첫번째값과 두번째 값을 매개변수로 받아 두 값을 비교하여 작은 값을 리턴하는 함수  int smaller(int first, int second)  {  return (first < second ? first : second);  }  //첫번째값과 두번째 값을 매개변수로 받아 두 값을 비교하여 큰 값을 리턴하는 함수  int larger(int first, int second)  {  return (first > second ? first : second);  }  ~  ~ |
| 실행결과 |
| 그래프C:\Users\kabsung\AppData\Local\Microsoft\Windows\INetCache\Content.Word\KakaoTalk_20190323_162946288.jpg |