20185237 류진영

#include <stdio.h>

int main()

{

int a[] = { 8,2,8,1,3 };

int \*p = a;

int ary[][4] = { 5,7,6,2,7,8,1,3 };

int(\*ptr)[4] = ary;

printf("%2d, %2d\n", \*(p+1), \*(p+4)); //p에 a배열의 주소값을 집어넣었다. a 배열의 1번과 4번 인덱스의 값이 출력됨 (2,3)

printf("%2d, %2d\n", p[0],p[4]); //p가 참조한 a배열의 0번과 4번 인덱스에 해당되는 값이 출력됨

printf("sizeof(a) = %d, sizeof(p) = %d\n",sizeof(a),sizeof(p));//a와 p 변수의 크기를 출력 p는 주소값을 담는 포인터 변수이니 4, a는 정수형 데이터 5개를 담은 정수형 배열이니 20

printf("%2d, %2d\n", \*\*ary, \*\*ptr++);

printf("%2d, %2d\n", \*\*(ary + 1), \*\*(ptr++));

ptr = ary;

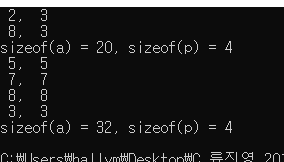
printf("%2d, %2d\n", \*(ary[1]+1), \*(ptr[1] +1));

printf("%2d, %2d\n", \*(\*(ary + 1) + 3), \*(\*(ptr + 1) + 3));

printf("sizeof(a) = %d, sizeof(p) = %d\n", sizeof(ary), sizeof(ptr));

return 0;

}



#include <stdio.h>

#define SIZE 3

int main()

{

int i, sum = 0;

int score[] = { 89,98,76 };

printf("%2s %10s %6s\n", "첨자","주소","저장값");

for (i = 0; i < SIZE; i++)

printf("%2d %10u %6d\n",i,(score+i), \*(score + i)); //순서대로 첨자, 주소(i번째 인덱스의 주소), 저장값(i번째 인덱스에 해당되는 값)을 출력한다.

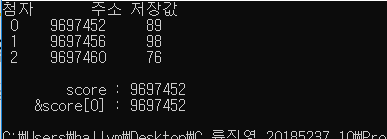
puts(""); //개행

printf("%15s %u\n","score :", score); //score의 주소 출력 (0번째 인덱스의 주소와 동일)

printf("%15s %u\n", "&score[0] :", &score[0]); //score[0]의 주소 출력 (0번째 인덱스의 주소)

return 0;

}



#include <stdio.h>

#define ROW 2

#define COL 3

int main()

{

int td[][COL] = { {8,5,4},{2,7,6} }; //2차원 배열 선언

int i = 0, j = 0, cnt = 0;

printf("%d, %d, %d\n", sizeof(td), sizeof(td[0]), sizeof(td[1])); //순서대로 td 배열 전체 크기, td 배열의 첫번째 행, td 배열의 두번째 행 크기 출력

printf("%u, %u, %u\n",td, td[0], td[1]); //순서대로 td 배열의 주소, td 배열의 첫번째 행, td 배열의 두번째 행 주소 출력 td와 td[0]의 주소값은 동일하다

printf("%u, %u, %u\n",td, td[0][0], td[1][0]); //td의 주소, td[0][0]의 값, td[1][0]의 값 출력

\*\*td = 10; //td[0][0] = 10

\*td[1] = 20; //td[1][0] = 10

for (i = 0; i < ROW; i++)

{

for (j = 0; j < COL; j++, cnt++)

{

printf("%d %d, ", \*(\*td + cnt), \*(td[i] + j)); //td[0][cnt]와 td[i][j] 출력, 이때 td[0][cnt]를 의미하는 \*(\*td + cnt) 같은 경우 td[0][0]에서 td[1][2]까지 주소값이 1씩 증가하게 되기 때문에 cnt가 3 이상이 될 경우 td[1][cnt-3]을 의미한다

}

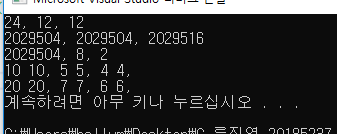
puts("");

}

system("pause");

return 0;

}



#include <stdio.h>

int main()

{

int ary[][5] = { 1,2,3,4,5,6,7,8,9,10 }, i,j; //{{1,2,3,4,5}, {6,7,8,9,10}}

int(\*ptr)[5]=ary;

printf("arr배열 출력\n");

for (i = 0; i < 2; i++)

{

for (j = 0; j < 5; j++)

{

printf("%-5d",ary[i][j]); //ary 배열 출력

}

puts(""); //개행

}

printf("배열 포인터를 사용하여 모든 원소의 값을 5 증가\n");

for(i=0; i<10; i++)

(\*ptr)[i] += 5;

printf("포인터만을 사용하여 출력\n");

for (i = 0; i < 10; i++)

{

printf("%-5d", \*\*ptr+i); //ptr이 참조한 배열 출력 (ary[0][0]을 참조한 \*\*ptr부터 시작해서 주소값을 i씩 증가시키며 하나씩 출력한다. 이때 i는 ary배열의 열을 의미하는데 ary배열의 열 이상으로 i값이 증가하면 그 다음 행의 첫번째 열의 주소값을 의미한다.)

if(i==4)

puts(""); //개행

}

puts(""); //개행

printf("인덱스만을 사용하여 출력 \n");

for (i = 0; i < 2; i++)

{

for (j = 0; j < 5; j++)

{

printf("%-5d", ary[i][j]); //ary 배열 출력

}

puts(""); //개행

}

printf("포인터와 인덱스를 같이 사용하여 출력\n");

for (i = 0; i < 10; i++)

{

printf("%-5d", (\*ptr)[i]); //\*\*ptr+i과 같은 의미

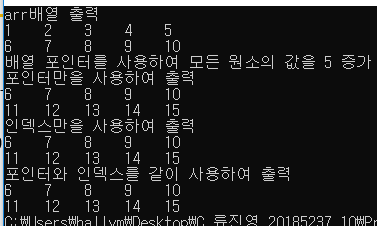
if (i == 4)

puts(""); //개행

}

return 0;

}



#include <stdio.h>

int main()

{

float a[][3] = { 4.2,4.3,3.8,3.7,1.5,0.7 }, b[][3] = { 5.2,2.1,1.8,3.7,1.5,0.7 };

int i, j;

float (\*ap)[3]=a, (\*bp)[3]=b; //a와 b 배열을 각 포인터 변수에 참조

printf("행렬 합\n");

for (i = 0; i < 6; i++)

{

printf("%-5.2f ", (\*ap)[i] + (\*bp)[i]); //순서대로 a[0][0]과 b[0][0] 부터 a[1][3] 과 b[1][3]을 더한 값을 출력한다.

if(i==2) //i가 2가 되었을 때 개행

puts("");

}

printf("\n행렬 차\n");

for (i = 0; i < 6; i++)

{

printf("%-5.2f ", (\*ap)[i] - (\*bp)[i]);//순서대로 a[0][0]과 b[0][0] 부터 a[1][3] 과 b[1][3]을 뺀 값을 출력한다.

if (i == 2) //i가 2가 되었을 때 개행

puts("");

}

return 0;

}

