

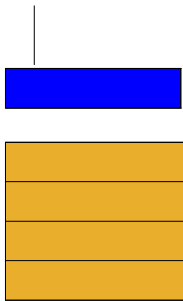
PART 16-2. 동적 할당 저장 공간의 활용

◆ 연습문제 ◆

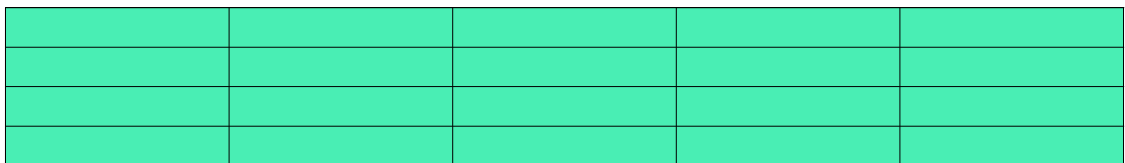
1. 4행 5열의 행렬의 값을 저장할 2차원 배열을 동적 할당하는 코드를 작성하세요.

2중 포인터

Matrix



동적 할당 공간



동적 할당 공간

정답 : `int **matrix = (int **) malloc(4 * sizeof(int *));`

```
for (i = 0; i < 4; i++) {  
    matrix[i] = (int *) malloc(5 * sizeof(int));  
}
```

해설 : 없음

2. 1번에서 만든 동적 할당 영역을 반환하는 코드를 작성합니다.

정답 : `for (i = 0; i < 4; i++) {
 free([matrix[i]]);`

```
}
```

```
free(matrix);
```

해설 : 없음

3. 다음은 명령행 인수를 사용하여 프로그램 이름을 제외한 나머지 인수를 출력하는 코드인데 빈칸을 채워보세요.

```
int main(          ①          ) {  
    int i;  
  
    for (i = 0; i < ( ② ); i++) {  
        printf("%s\n", ( ③ ));  
    }  
}
```

정답 : ① int argc, char **argv
 ② argc - 1
 ③ argv[i + 1]

해설 : 없음

4. 다음과 같이 행렬의 합을 구하는 프로그램을 동적 할당을 사용하여 작성하세요.

1	2	3	4
5	6	7	8
9	10	11	12

+

12	11	9	8
8	7	6	5
4	3	2	1

=

13	13	13	13
13	13	13	13
13	13	13	13

실행 결과

13 13 13 13

13 13 13 13

13 13 13 13

정답

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main(void) {
```

```
    int *pa[3], *pb[3], *pc[3];
```

```
    int i, j;
```

```
    for (i = 0; i < 3; i++) {
```

```
        pa[i] = (int *) malloc(4 * sizeof(int));
```

```
        pb[i] = (int *) malloc(4 * sizeof(int));
```

```
        pc[i] = (int *) malloc(4 * sizeof(int));
```

```
    }
```

```
    for (i = 0; i < 3; i++) {
```

```
        for (j = 0; j < 4; j++) {
```

```
            pa[i][j] = i * 4 + j + 1;
```

```

        pb[i][j] = 12 - ( i * 4 + j);
        pc[i][j] = pa[i][j] + pb[i][j];
    }
}

for (i = 0; i < 3; i++) {
    for (j = 0; j < 4; j++) {
        printf("%5d", pc[i][j]);
    }
    printf("\n");
}

return 0;
}

```

해설 : 없음

5. 명령행 인수를 통해 문자열을 반복적으로 출력하는 프로그램을 작성하세요. 프로그램의 이름은 print_string으로 하며, 실행할 때 출력할 문자열과 반복 횟수를 입력하면 반복 횟수만큼 문자열을 출력합니다.

[실행 결과 1]

```

C:\Wprint_string thanks 3
thanks
thanks
thanks

```

[실행 결과 2]

```

C:\Wprint_string thanks
프로그램 사용법 오류입니다.
사용법 : print_string 문자열 반복횟수

```

정답

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char ** argv) {
    int i;
    int count;

    if (argc < 3) {
        printf("프로그램 사용법 오류입니다.\n");
        printf("사용법 : string_print 문자열 반복횟수\n");
        exit(1);
    }

    count = atoi(argv[2]);

    for (i = 0; i < count; i++) {
        printf("%s\n", argv[1]);
    }

    return 0;
}
```

해설 : 없음