

3. 동적 메모리 할당

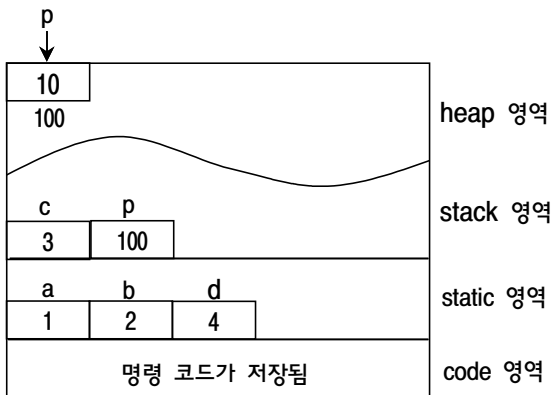
동적 메모리 할당은 실행시간에 힙 영역에 필요한 만큼 메모리를 할당하는 것이다.
 자료구조 시험에서 출제되는 C 언어 함수는 malloc(size_t size); 이다.

◆ 함수 malloc() 사용 - 원형 : void *malloc(size_t size);

<pre>char *p1; p1 = (char *)malloc(50); p1 = "KOREA"; printf("%s\n", p1); //출력 KOREA</pre>	<pre>int *p2; p2 = (int *)malloc(50 * sizeof(int)); p2[0] = 88; printf("%d\n", p2[0]); //출력 88</pre>
--	--

- 함수 malloc(size_t size)에서 할당하는 메모리 크기는 바이트 수이다.

```
// 변수의 기억장소 할당
int a = 1;
static int b = 2;
void main()
{
    int c = 3;
    static int d = 4;
    int *p; //포인터 선언
    p = (int *) malloc(sizeof(int)); //동적 메모리 할당 (힙 할당)
    *p = a + b + c + d;
    printf("%d\n", *p); //1 + 2 + 3 + 4 = 10
    free(p); //메모리 회수
}
```



- 변수 `p`는 스택 영역에 할당된다.

기출문제 분석

1. 다음은 동적 메모리 할당을 통해 구조체 공간을 확보하고 이름과 학번을 입력받아 구조체에 저장하는 C 언어 함수이다. ㉠, ㉡에 들어갈 내용으로 옳게 짝지어진 것은? [2016년 국가 7급]

```
#include <stdio.h>
#include <stdlib.h>
struct student
{
    char name[20];
    int id;
};
struct student *alloc()
{
    struct student *p;
    p = ㉠ ;
    if(p==NULL)
    {
        printf("Memory allocation error.");
        return p;
    }
    printf("Student name?\n");
    gets(p->name);
    printf("Student id?\n");
    scanf("%d", ㉡ );
    return p;
}
```

- | | |
|--|------------|
| ㉠ | ㉡ |
| ① (struct student)malloc(sizeof(struct student)) | &((*p).id) |
| ② (struct student *)malloc(sizeof(struct student)) | p->id |
| ③ (struct student)malloc(sizeof(struct student)) | (*p).id |
| ④ (struct student *)malloc(sizeof(struct student)) | &(p->id) |

☞ 구조체 포인터

```

// 프로그램을 완성시켜 보면
#include <stdio.h>
#include <stdlib.h>
struct student{
    char name[20];           //배열명 name은 배열의 시작주소를 가진다.
    int id;
};
struct student *alloc() {
    struct student *p;           //구조체 포인터 선언
    p = (struct student *)malloc(sizeof(struct student)); //동적 메모리 할당 (힙 할당)
    if(p==NULL){
        printf("Memory allocation error.");
        return p;
    }
    printf("Student name?\n");
    gets(p->name);           //name 입력
    printf("Student id?\n");
    scanf("%d", &(p->id));    //id 입력, scanf()에서 인수는 반드시 주소, &가 필요
    printf("Student name : %s \n", p->name); //name 출력
    printf("Student id : %d \n", p->id);    //id 출력, &가 필요 없다.
    return p;
}
void main(){ alloc(); }

// 만약, 구조체가 다음처럼 정의되어 있으면
struct student
{
    char name[20]; //배열명 name은 배열의 시작주소를 가진다.
    char id[20];  //id도 char형 배열로 선언되어 있다
};

↓
scanf("%s", p->id); //id 입력, &가 필요 없다. id가 배열명으로 주소를 가지므로

```