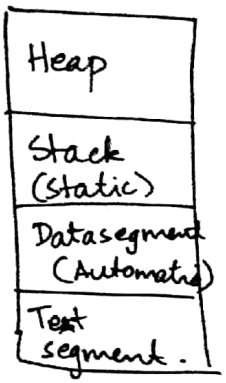
Dynamic Memory Allocation

Allocation of memory during runtime or execution time is called dynamic memory allocation.

In dynamic memory allocation, the memory will be allocated (for the calling function) (from the region called heap or store)



Q. Compare static with dynamic memory allocation.

Functions used to perform dynamic memory allocation are as follows:-

1. malloc
2. calloc
3. realloc

} allocation.

4. free } deallocation.

Dagling pointer → one that points to garbage address

malloc

- It allocates single block of memory
- On failure, it returns NULL
- On success, it returns the starting address (pointer) from where memory is allocated.
- The return type of the malloc function is a void pointer. which can be type casted based on the purpose
- By default, garbage values will be stored under the allocated memory

Syntax

```
#include <stdlib.h>

void * malloc (int size)   /* one argument for mallac func*/
```

-

```
datatype *ptr;
    ptr= (datatype*) malloc (sizeof (datatype));
  if (ptr==NULL)
  {
  pf("insufficient memory");
     exit(0);
   }
```

Write the syntax to allocate memory dynamically for the following:-

i) for 1 integer.

```
int *ptr;
ptr = (int *) malloc (sizeof(int));
```

ii) for n integers.

```
int *ptr , n;
ptr = (int *) malloc (n * sizeof (int));
```

iii) for n characters

```
char *ptr;
int n;
ptr = (char *) malloc(n*size of (char));
```

WAP to find Σ^{n} elements ~~of array~~ using dynamic memory allocation.

```
int main()
{
    int *ptr, i, sum=0, n;
    pf("\n Read value for n: ");
    sf("%d ", &n);
    ptr = (int *) malloc(n * sizeof (int));
    pf("Reand n elements");
    for (i=0; i<n; i++)
    {
        sf("%d", ptr+i);
        sum = sum + *(ptr+i);
    }
```

WAP to read and print one student info using DMA.

```
Struct student
{
    char name[10];
    int sem;
}
int main()
{
    struct student *ptr;
    ptr = (struct student *) malloc (sizeof (struct student));
    pf("Read name");
    sf("%s", ptr->name);
    pf("Read sem");
    sf("%d", &ptr->sem);
}  pf...
```

WAP to read n students...

```
struct student
{
    char name[10];
    int sem;
}
int main()
{
    struct student *ptr;
    int n, i;
    pf(Read n); sf(n);
    ptr = (struct student *) malloc(n * sizeof(struct student));
```

calloc

Allocates memory dynamically using multiple blocks return type is void pointer which can be typecasted.

Syntax:

```
void * calloc (int n, int size)
```

first argument n is no of blocks
2nd arg size is size of each block.

By default, a value of arithmetic zero will be stored.

Realloc used to resize (increase or decrease) the memory allocated either by malloc or calloc

SYNTAX:

```
void * realloc (void *ptr, int newsize)
```

Free deallocates the memory allocated previously either by malloc, calloc, or realloc.

SYNTAX:

```
void free (void *ptr)
```