Dynamic memory Allocation

Dynamic memory allocation in c language enables the c programmer to allocate memory at runtime Dynamic memory allocation in c language is possible by 4 functions of stdlib.h header file

1) malloc ( ) - Used for allocating the memory
2) calloc ( ) - For allocating the memory
3) realloc ( ) - For updating the memory
4) free ( ) - Deleting the memory

Difference between static and dynamic memory allocation

static memory allocation	dynamic memory allocation
1) memory is allocated at compile time.	1) memory is allocated at run time
2) memory can't be increased while executing program	2) memory can be increased while executing program.
3) used in array	3) used in linkedlist.

Syntax

malloc:

```
int * p = (int *) malloc (n * size of(int));
```

calloc:

```
int * p = (int *) calloc (n, size of (int));
```

free:

```
free (*p);
```

Malloc ( )	calloc ( )
Malloc() is DMA function used for allocating memory	calloc ( ) is DMA function used for allocation memory syntax is
int * p = int * (malloc) (n * size of (int));	int * p = (int *) calloc
Memory is allocated in single block.	Memory is allocated in different block.
Default value is garbage Value	Here default value is zero

Dynamic memory allocation

```
// Example For malloc()
 // malloc ()
  # include <stdio.h>
  # include <Stdlib.h>
   int main()
  { int * p;
    int i, n;
    printf ("Enter the range \n");
    scanf ("%d", &n);
    p - (int *) malloc (n * sizeof (int));
    for (i = 0; i < n; i++)
{ p[i] = i+1;
}
printf("The value assigned are\n");
for (i=0; i<n; i++)
  printf(" %d\n", (* p+i));
}

//malloc

 #include <stdio.h>
 #include <stdlib.h>
  int main ()
  {
   int n, i, *ptr, sum=0;
   printf("Enter number of elements: ");
   scanf("%d ", &n);
    ptr=(int*) malloc (n* sizeof (int));
    if(ptr==NULL)
    {printf ("Error ! memory not allocated ");
     exit(0);
    }
     printf("Enter elements of array ");
     for (i=0 ; i<n; i++)
      { scanf("%d", ptr+i);
        sum+=*( ptr + i);
     }
      printf("sum of = %d", sum);
    } free (ptr);
    } return 0;
```

```
// calloc
# include <stdio.h>
# include <stdlib.h>
int main()
{
int *p;
int i,n;
printf("Enter the range\n");
scanf("%d", &n);
p=(int *)calloc(n, size of(int));
for (i=0; i<n; i++)
{
p[i] = i+1;
}
printf(" the value assigned are \n");
for(i=0; i<n; i++)
printf("%d\n", *(p+i));
}
```

```
# include <stdio.h>
# include <stdlib.h>
 int main ()
 {
  int n, i ,*ptr, sum=0;
 printf ("Enter the number of elements: ");
 scanf ("%d", &n);
 ptr = (int*) calloc (n, sizeof(int));
  if (ptr==NULL)
   {
    printf (" Error! memory not allocated.");
  exit(0);
   }
   printf (" Enter Elements of array: ");
   for (i=0; i<n; i++)
  {
   scanf ("%d", ptr+i);
    sum+=*(ptr+i)
   }
   printf ("sum%d", sum);
   free (ptr);
   return0;
   }
```

```
# include <stdio.h>
# include <stdlib.h>
# include <string.h>
  int main()
{
 char * p1
  int m1, m2;
  m1 = 15;
 m2 = 20
  p1 = (char*) malloc(m1);
 strcpy(P1, "Mahatma Gandhi");
  P1 = (char*) realloc (P1, m2);
 strcat(P1 , "Institute of Technology");
  printf("%s\n", P1);
   return 0;
}
```