Swati Mall

Computer Science and Engineering

II Semester

Pointers to Functions :

A particular confusing yet powerful feature of C is the "function pointer".

A function has a physical location in memory that can be assigned to a pointer. This address is the entry point of the function and it is the address used when the function is called.

Example: WAP to reverse a string.

```
#include<stdio.h>
#include<conio.h>
char *reverse(char *);
void main()
{
 clrscr();
 printf("%s", reverse("Computer"));
getch();
}
char *reverse(char *p)
 {
  int l, i;
  char t;
for(l=0; *(p+l)!='\0'; l++)
for(i=0; i<2; i++)
```

```
{
  t = *(p+i);
  *(p+i) = *(p+d-1-i);
  *(p+d-1-i) = t;
}
```

Output: retupmoc

Dynamically Allocated Arrays

Dynamic allocation is means by which a program can obtain memory while it is running. Global variables are allocated storage at compile time. Non-static, local variable use the stack.

The core of C's allocation system consist of the functions malloc(), calloc(), realloc() and free()

* malloc() — It is used to allocate the space in memory at the time of execution of program.
  → It carry garbage value.
  → It store single argument.

Syntax: malloc(n* sizeof(int));

* calloc( ) — It is used in allocation of space in memory at the time of execution of program.
→ It initialize the allocated memory from zero.
→ It takes two arguments.

Syntax: calloc(n, size of(int));

* realloc( ) — It is used to modify the allocated memory.
→ It deallocate the old object and allocate the new object.
→ It takes two arguments.

Syntax: realloc(ptr, n*size of(int));

* free( ) — This function is used to dynamically deallocate the memory.
The memory allocated by calloc( ) and malloc( ) is doesn't deallocate by itself. We have to deallocate the memory by using free( ) function.

Syntax: free(*ptr)