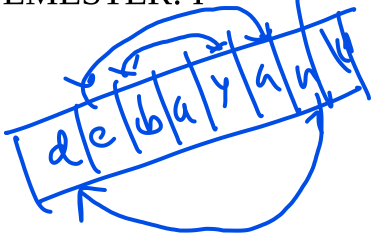
LAB-ASSIGNMENT 4 (String)

SUBJECT: Computer Lab. SUBJECT CODE: CS111. SEMESTER: I



madam

1. ✓ Write a program to print a string. ✓
2. Write a program to find the length of a string:
   a. Using 'strlen' function ✓
   b. Without using 'strlen' function ✓
   c. Using a character pointer ✓
3. Write a program to concate two strings:
   d. Using 'strcat' function
   e. Without using 'strcat' function
4. ✓ Write a program to reverse a string. ✓
5. Write a program to check whether a string is palindrome or not.
6. Write a program to copy source string to destination string.
   a. Using 'strcpy' function. ✓
   b. Without using 'strcpy' function. ✓
7. ✓ Write a program to compare two strings. ✓

H.W "I love to read books"

O/p:- "I evol ot daer skoob"

O/p:- "books read to love I"

2nd

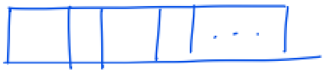
4 ~~dig~~ digits given by user :- 4320

O/p:- four thousand three hundred twenty.

enter a +ve 4digit no and display it in words

(Runtime) Dynamic memory allocation. ??

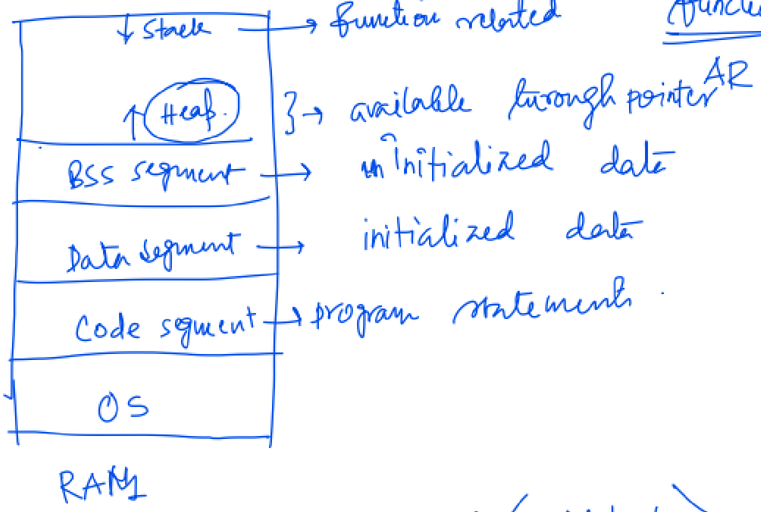
int a[50];



function AR

```
main()
{ int a;
  main();
}
```

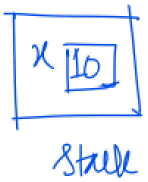
o/p: stack overflow


↓ Stack
→ function related
↑ Heap
} → available through pointer
BSS segment → uninitialized data
Data segment → initialized data
Code segment → program statements.
OS
RAM

Malloc(), calloc(), realloc(), free() :- <stdlib.h>

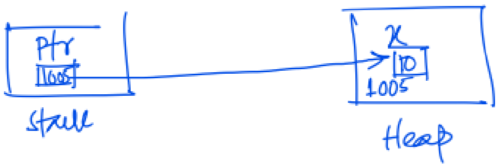
static (compilation time)

int x = 10; 4 bytes of memory is allocated at compilation time and the value 10 is stored. This mym is allocated on the "stack"



Static m/m allocation

⊛ Static m/m allocation improves the performance of the program by faster access of the data.



⊛ dynamic m/m allocation makes the program execution slower.

⊛ While allocating the dynamic m/m the programmer should explicitly use functions like malloc(), calloc() and realloc() to allocate m/m. on heap.

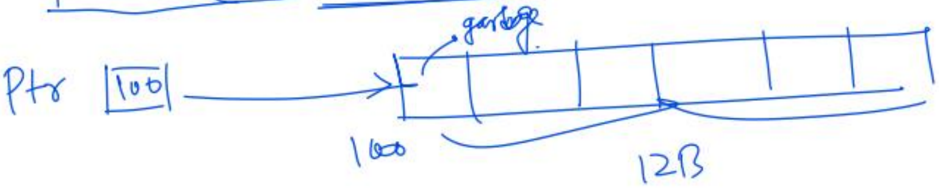
Once m/m usage is completed, the programmer should free the memory using free() function.

malloc

```
(void *) malloc (size_t size);
```

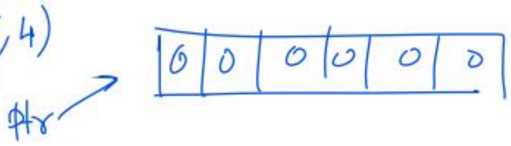
argument specifies the no of bytes to be allocated.

```
int *ptr;
ptr = (int *) malloc (12);
```



A void pointer is a pointer that can be converted into any type of pointer.

```
ptr = (int *) calloc (5,4)
```



String.

'A' → character

"A" → String

A	\0

str[] = "debayan"

str

d	e	b	a	y	a	n	\0
100	101	102	103	104	105	106	107

strlen () :- library function

```
#include <stdio.h>
#include <string.h>
int main ()
{ char str[50];
  printf ("enter a string");
  gets (str);
  printf ("Length of the string is : %u", strlen(str));
      return 0;
  }
```

Array version

```
unsigned int astrlen (char a[])
    {   int i;
        for (i=0; a[i] != '\0' ; i++);
            return i;
    }
```

pointer :-

```
unsigned int pastrlen (char *str)
    {char *p = str;
      while (*p != '\0')
            p++;
          return p-str;
    }
```