18-07-2024

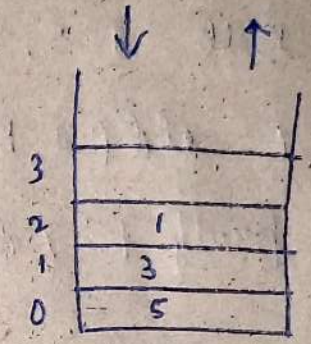
STACK :- It is a linear Data Structure. The data is inserted from TOP end & deleted from the same end.

It operates on the principle of LIFO.

Top is a pointer which always points top of the stack

STACK is "finite"

* overflow * underflow



OPERATIONS :=

-> Push

-> Pop

-> Traverse

TOP element address ← 2 TOP

Ex:

define :- It is a macro, which substitute the code.

-> # define add(a,b) a+b

main ()

{

int a = 5, b = 6, c;

c = add (a, b);

printf ("%d", c);

}

o/p :- 11

typedef :- It is used to rename a datatype

-> # define add(a, b) a+b

typedef int sree;

main()

{

sree a=5, b=6, c;

c = add(a, b);

printf("%d", c);

}

o/p :- 11

-> # define max 5

int s[max], top = -1;

5 -> same size for max

void push()

{

int elem;

if (top == max - 1)

{

printf("overflow");

return;

}

printf("enter elements");

scanf("%d", &elem);

top++;

s[top] = elem;

}

stack.c

```
void pop()
{
  int k;
  if ( top == -1)
  {
    printf (" underflow ");
    return;
  }
  k = s[ top ];
  top--;
  printf (" del element is %d ", k);
}
void display ( )
{
   int i;
   for( i=0; i<= top ; i++)
   printf (" %d ", s[i]);
}
main ( )
{
   int ch;
   while (1)
   {
      printf (" Enter 1 for push 2 for pop 3 for
        display 4 for exit ");
      scanf ("%d", &ch);
```

```
switch (ch)
{
    case 1: push();  break;
    case 2: pop();  break;
    case 3: display();  break;
    case 4: exit(0);
}
}
}
```