GAYATHRI

4AL19CS035

E - SECTION

Algorithm

Step1: Start
Step2: void push (int *stack, int data);
Step3: void pop (int *stack)
Step4: void display (int *stack)

Step6: top = -1
Step7: Display Enter the size of the stack
input size
Step8: stack = (int *)malloc (sizeof (int) * size);
Step9: Display Stack operation using array
Step10: Display 1. PUSH 2. POP 3. DISPLAY 4. EXIT
Step11: do
Display Enter the choice
input ch
Switch (ch)
case 1: Display Enter your i/p to push
input data
push (stack, data)
break
case 2: pop (stack)
break
case 3: display (stack)
break
case 4: Display EXIT POINT.
break;
default: Display Please Enter a valid
Choose (1/2/3/4)

while (ch != 4)
return 0
Step12: Stop

void push(int * Stack , int data)

Step 1: Entry
Step 2: if (top >= size - 1)
        Display Stack overflow
        return
Step 3: top++
Step 4: Stack[top] = data
Step 5: Display pushed +.d into the stack
        output Stack [top]
Step 7: return

void pop (int * Stack)

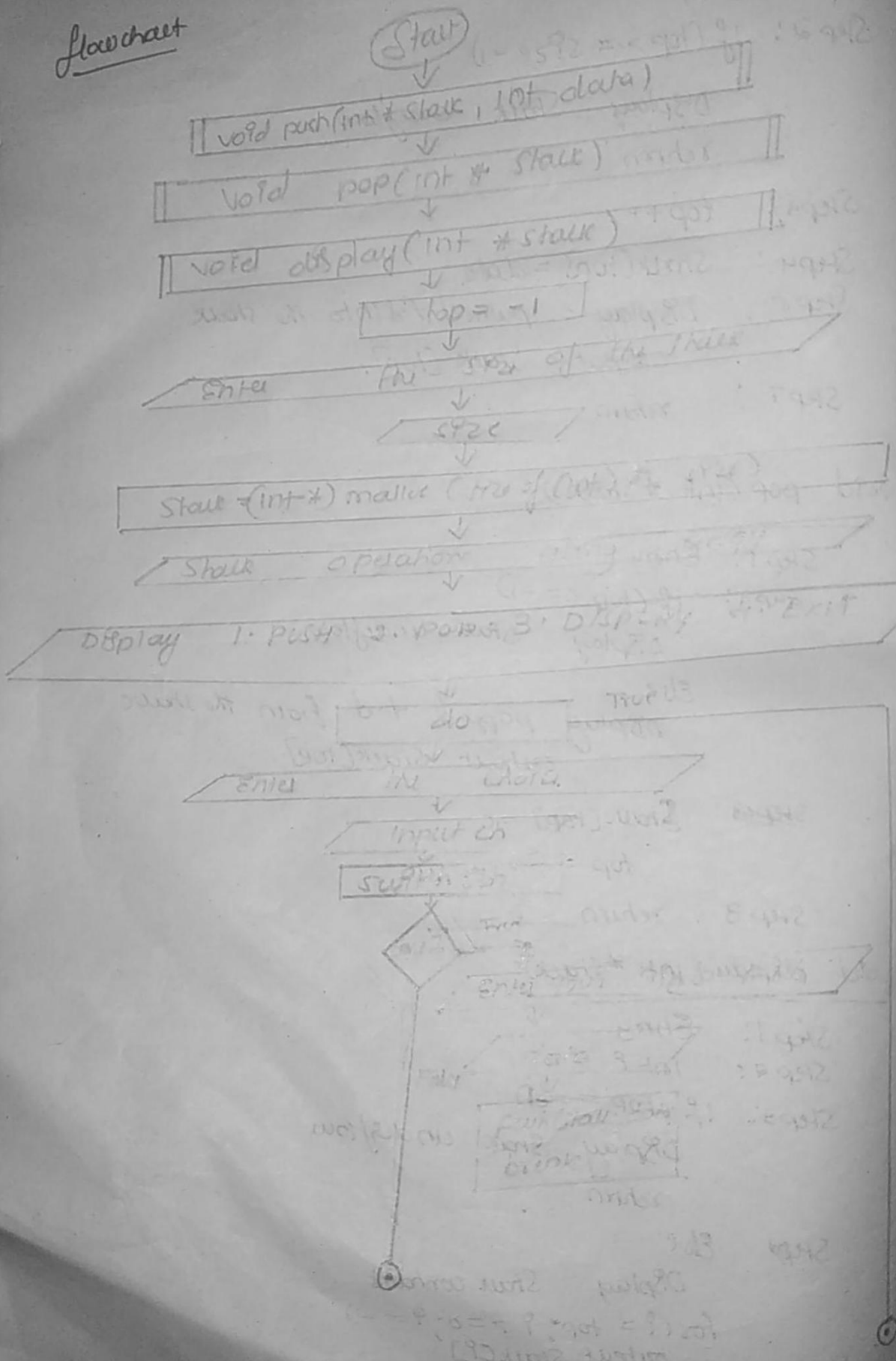
Step 1: Entry
Step 2: if (top <= -1)
        Display Stack underflow
        Else
        Display popped +.d from the stack
        output Stack[top]

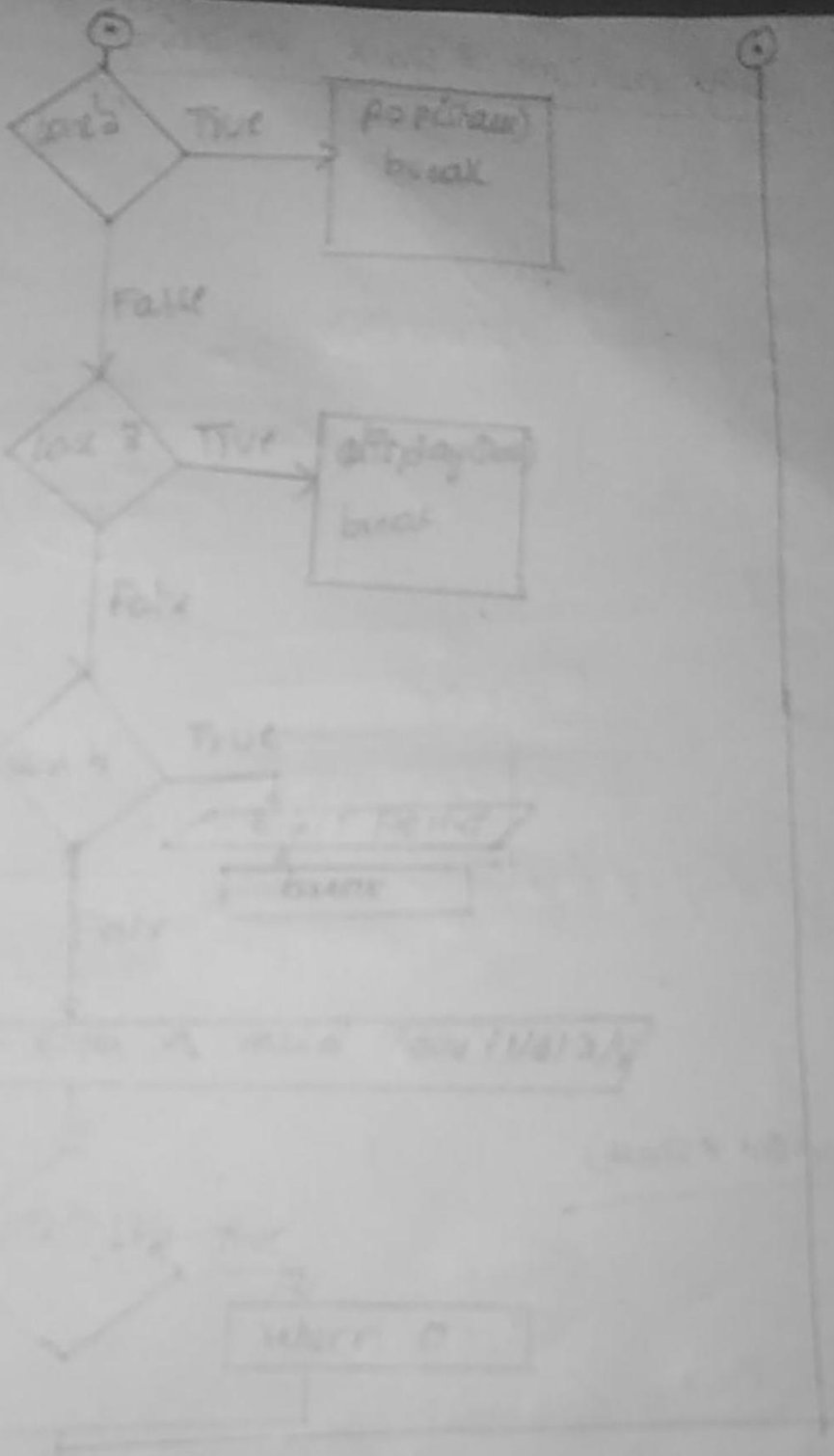
Step 3: Stack[top] = 0
        top - -
Step 3: return

void display (int *stack)

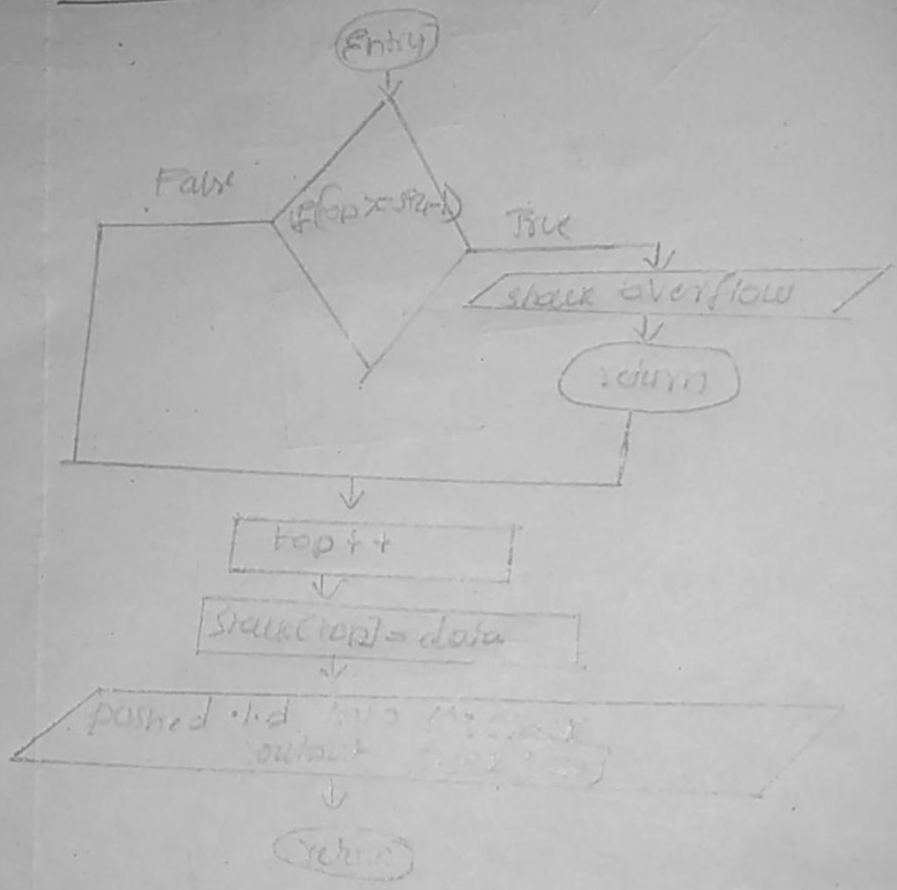
Step 1: Entry
Step 2: int i
Step 3: if (top == -1)
        Display Stack underflow
        return
        Else
        Display Stack contains
        for (i = top; i >= 0; i - -)
        output Stack[i]
Step 4: return

Flowchart

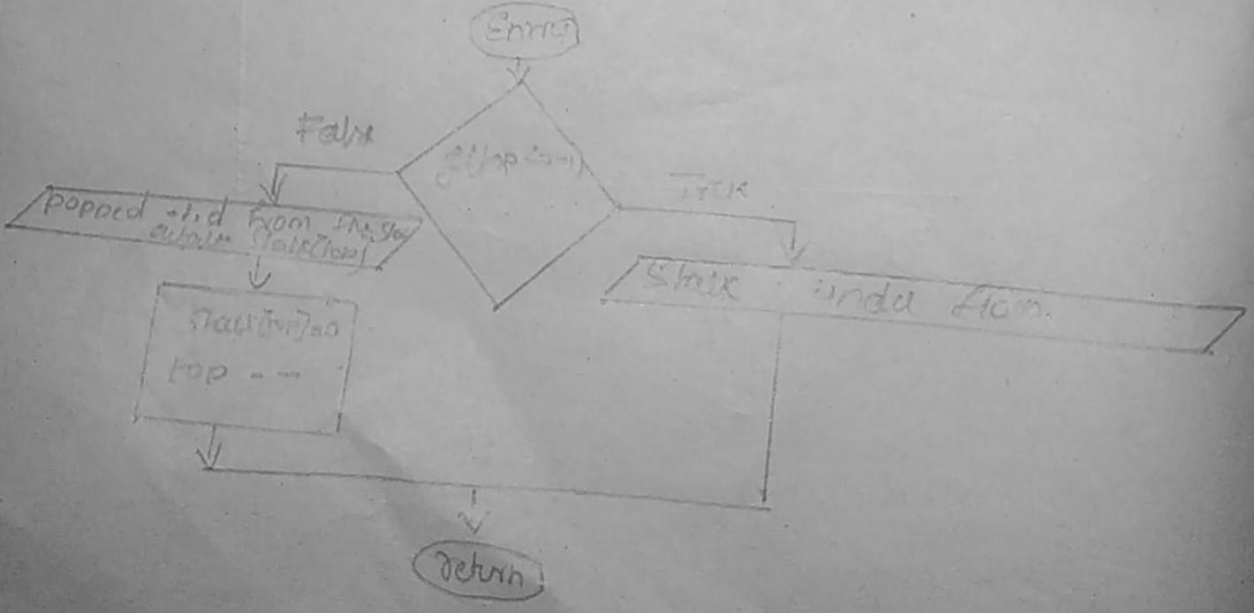

Start
void push(int * stack, int data)
Void pop(int * stack)
void display(int * stack)
top = -1
Enter the size of the stack
size
stack = (int *) malloc (
stack operation
display 1. PUSH
do
Enter the data
Input ch
switch


True
False
True
False
True

void push(int *Stack, int data)



void pop(int *Stack)



void display (int * stack)