2-Stack ques :

#include <stdio.h>

#define SIZE 100

int s[SIZE];

int top = -1;

void push(int value) {

if (top == SIZE - 1) {

printf("It is overflow\n");

} else {

top++;

s[top] = value;

}

}

void pop() {

if (top == -1) {

printf("Stack is underflow\n");

} else {

printf("Popped num is: %d\n", s[top]);

top--;

}

}

void display() {

if (top == -1) {

printf("Stack underflow\n");

} else {

printf("Stack of the inputs is: ");

for (int i = top; i >= 0; i--) {

printf("\n%d", s[i]);

}

printf("\n");

}

}

int main() {

push(78);

push(23);

push(65);

push(75);

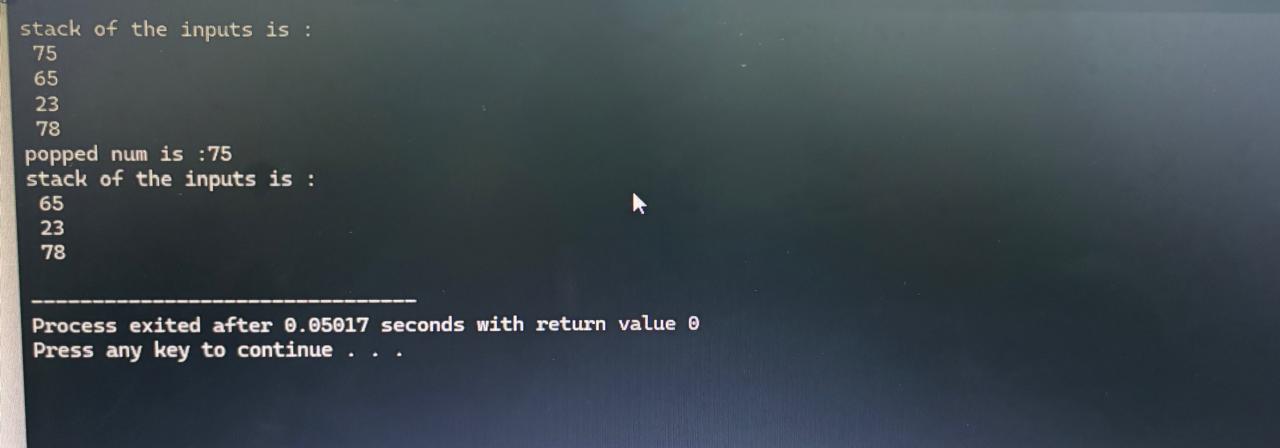
display();

pop();

display();

return 0;

}



1-ques tower of Hanoi:

#include <stdio.h>

#include <math.h>

void moveDisk(char from, char to, int disk) {

printf("Move disk %d from %c to %c\n", disk, from, to);

}

int main() {

int n;

printf("number of disks: ");

scanf("%d", &n);

char src = 'A', aux = 'B', dest = 'C';

if (n % 2 == 0) {

char temp = dest;

dest = aux;

aux = temp;

}

int totalm = pow(2, n) - 1;

for (int i = 1; i <= totalm; i++) {

if (i % 3 == 1) {

if (i % 2 == 1) moveDisk(src, dest, 1);

} else if (i % 3 == 2) {

if (i % 2 == 1) moveDisk(src, aux, 1);

} else if (i % 3 == 0) {

if (i % 2 == 1) moveDisk(aux, dest, 1);

}

}

return 0;

}

