Implement any one storage allocation strategies (heap, stack, static)

Aim –

To implement stack storage allocation strategy.

Code –

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#define size 5

struct stack{

int s[size];

int top;

}st;

int stfull(){

if(st.top >= size - 1)

return 1;

else

return 0;

}

void push(int item){

st.top++;

st.s[st.top] = item;

}

int stempty(){

if(st.top == -1)

return 1;

else

return 0;

}

int pop(){

int item;

item = st.s[st.top];

st.top--;

return (item);

}

void display(){

int i;

if(stempty())

printf("\nStack is Empty");

else{

for(i = st.top; i >= 0; i--)

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

}

}

int main(){

int item, choice;

char ans;

st.top = -1;

printf("\nImplementation of Stack");

printf("\nMain Menu");

printf("\n1.Push \n2.Pop \n3.Display \n4.exit"); do{

printf("\nEnter your Choice ");

scanf("%d", &choice);

switch(choice){

case 1:

printf("\nEnter the item to be pushed ");

scanf("%d", &item);

if(stfull())

printf("\nStack is full");

else

push(item);

break;

case 2:

if(stempty())

printf("\nEmpty Stack");

else{

item = pop();

printf("\nThe popped item is %d",

item);

}

break;

case 3:

display();

break;

case 4:

goto halt;

}

printf("\nDo you want to continue?");

ans = getch();

}

while (ans == 'Y' || ans == 'y');

halt:

return 0;

}

Output –



Result –

Implementation of stack storage allocation strategy is performed.

----------------X----------------