Data Structures:

Stack Operations:

1)#include<stdio.h>

int main()

{

int ch;

do

{

printf("\n stack operations\n");

printf("\n 1.push");

printf("\n 2.pop");

printf("\n 3.display");

printf("\n 4.peek");

printf("\n 5.exit");

printf("\nenter your choice");

scanf("%d",&ch);

}

while(ch!=5);

return 0;

}

2)#include<stdio.h>

int main()

{

int ch;

do

{

printf("\n stack operations\n");

printf("\n 1.push");

printf("\n 2.pop");

printf("\n 3.display");

printf("\n 4.peek");

printf("\n 5.exit");

printf("\nenter your choice");

scanf("%d",&ch);

}

while(ch!=5);

return 0;

}

3)#include<stdio.h>

#include<stdlib.h>

void push();

void pop();

void peek();

void display();

#define max 5

int a[max];

int ele;

int top=-1;

int main()

{

int ch;

do

{

printf("\n stack operations\n");

printf("\n 1.push");

printf("\n 2.pop");

printf("\n 3.display");

printf("\n 4.peek");

printf("\n 5.exit");

printf("\nenter your choice");

scanf("%d",&ch);

switch (ch)

{

case 1:

push();

break;

case 2:

pop();

break;

case 3:

display();

break;

case 4:

peek();

break;

case 5:

exit(0);

break;

}

}

while(ch!=5);

return 0;

}

void push()

{

if(top==max-1)

{

printf("\n stack is full");

return ;

}

else

{

printf("enter the element\n");

scanf("%d",&ele);

top++;

a[top]=ele;

printf("%d is inserted",ele);

}

}

void pop()

{

if(top==-1)

{

printf("\n stack is empty");

return ;

}

else

{

ele=a[top];

top--;

printf("%d is deleted",ele);

}

}

void display()

{

int i;

if(top==-1)

{

printf("\n stack is empty");

return ;

}

else

{

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

printf("%d ",a[i]);

}

}

void peek()

{

if(top==-1)

{

printf("\n stack is empty");

return ;

}

else

printf("%d is ready to delete",a[top]);

}