**STACK**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//Khai bao 1 Stack

typedef struct{

int Data[100];

int top;

} Stack;

//Tao rong

void MakenullStack(Stack \*S){

S->top=100;

}

//Kiem tra rong

int EmtyStack(Stack S){

if (S.top==100) return 1;

return 0;

}

//Kiem tra day

int FullStack(Stack S){

if (S.top==0) return 1;

return 0;

}

//Lay gia tri dinh

int Top(Stack S){

if (EmtyStack(S)!=1)

{

return S.Data[S.top];

}

}

//Xoa gia tri dinh

void Pop(Stack \*S){

if (EmtyStack(\*S)!=1)

{

S->top=S->top+1;

}

}

//Them vao gia tri tai dinh

void Push(int x, Stack \*S){

if (FullStack(\*S)==0)

{

S->top=S->top-1;

S->Data[S->top]=x;

}

else printf ("Loi day!!\n");

}

//Tim phan tu co trong x khong?

int Search(int x, Stack S){

for (int i=0;i<S.top;i++)

{

if(S.Data[i]==x) return 1;

}

return 0;

}

//Chuyen sang so nhi phan

Stack Binary(int x){

Stack S;

MakenullStack(&S);

int Bin;

while (x!=0)

{

Bin=x%2;

Push(Bin,&S);

x=x/2;

}

return S;

}

//Chuyen sang so bat phan

Stack Octal(int x){

Stack S;

MakenullStack(&S);

int Oct;

while (x!=0)

{

Oct=x%8;

Push(Oct,&S);

x=x/8;

}

return S;

}

//Hien thi

void hienthi(Stack \*pS){

while (EmtyStack(\*pS)!=1)

{

printf("%d ",pS->Data[pS->top]);

Pop(pS);

}

}

int main(){

Stack S;

MakenullStack(&S);

int n,x;

printf ("\t- Nhap vao so phan tu toi da co trong Stack:");

scanf ("%d",&n);

printf("\n");

for (int i=0;i < n;i++)

{

printf("\t\t+ Nhap gia tri phan tu thu %d:",i+1);

scanf ("%d",&x);

Push(x,&S);

}

Stack T;

MakenullStack(&T);

for (int j=0; j<n;j++)

{

Push(Top(S),&T);

Pop(&S);

}

printf("\t- Hien thi cac phan tu co trong danh sach:");

hienthi(&T);

printf("\n");

int t;

printf("\t- Nhap sao can tim:");

scanf ("%d",&t);

if(Search(t,S)==1)

{

printf("\t---> So %d co trong Stack\n",t);

}

else

{

printf("\t--->So %d khong co trong Stack\n",t);

}

int k;

Stack K;

MakenullStack(&K);

printf("\t- Nhap so can chuy doi:");

scanf("%d",&k);

printf("\t- So %d sau khi chuyen sang:\n",k);

printf("\t\t+ Nhi phan: ");

K=Binary(k);

hienthi(&K);

printf("\n");

printf("\t\t+ Bat phan: ");

K=Octal(k);

hienthi(&K);

}