C C D B

O(n^2) O(n\*m) O(log2n)

C C B B C D C

**15利用栈实现十进制到二进制的转换（利用除2取余法）**

**输入：45**

**输出：10 1101**

#include<stdio.h>

#include<malloc.h>

#include<math.h>

#include<string.h>

#include "process.h"

#define SIZE 100

#define STACKINCREMENT 10

#define OK 1

#define ERROR 0

#define TRUE 1

#define FALSE 0

typedef int Status;

typedef struct

{

int a;

} SElemType;

typedef struct

{

SElemType \*base;

SElemType \*top;

int stacksize;

} SqStack;

SqStack S; //定义全局变量

Status InitStack(SqStack \*S)

{

S->base=(SElemType \*)malloc(SIZE\*sizeof(SElemType));

if(!S->base) exit(OVERFLOW);

S->top=S->base;

S->stacksize=SIZE;

return OK;

}

Status Push(SqStack \*S,SElemType e)

{

if(S->top-S->base>=S->stacksize)

{

S->base=(SElemType \*)malloc((S->stacksize+STACKINCREMENT)\*sizeof(SElemType));

if(!S->base) exit(OVERFLOW);

S->top=S->base+S->stacksize;

S->stacksize+=STACKINCREMENT;

}

\*S->top++=e;

//printf("%dwww\n",\*--S->top);

return OK;

}

Status Stackempty(SqStack \*S)

{

if(S->top==S->base)

return TRUE;

else

return FALSE;

}

Status Pop(SqStack \*S,SElemType \*e)

{

if(S->top==S->base) return ERROR;

\*e=\*--S->top;

return OK;

}

Status DtoBTrans(int N,SqStack \*S)

{

SElemType e;

while(N)

{

e.a=N%2;

Push(S,e);

N=N/2;

}

while(!Stackempty(S))

{

Pop(S,&e);

printf("%d",e);

}

return OK;

}

void main()

{

int x;

InitStack(&S);

printf("请输入十进制数：");

scanf("%d",&x);

DtoBTrans(x,&S);

}

**16.以定长顺序存储结构表示串，求S串和T串的最长公共子串**

#include <stdio.h>

#include <malloc.h>

#include <string.h>

char \*commanstring(char shortstring[], char longstring[])

{

int i, j;

char \*substring=malloc(256);

if(strstr(longstring, shortstring)!=NULL) //如果……，那么返回shortstring

return shortstring;

for(i=strlen(shortstring)-1;i>0; i--) //否则，开始循环计算

{

for(j=0; j<=strlen(shortstring)-i; j++)

{

memcpy(substring, &shortstring[j], i);

substring[i]='\0';

if(strstr(longstring, substring)!=NULL)

return substring;

}

}

return NULL;

}

main()

{

char \*str1=malloc(256);

char \*str2=malloc(256);

char \*comman=NULL;

gets(str1);

gets(str2);

if(strlen(str1)>strlen(str2)) //将短的字符串放前面

comman=commanstring(str2, str1);

else

comman=commanstring(str1, str2);

printf("the longest comman string is: %s\n", comman);

free(str1);

free(str2);

}

**17.循环队列的应用——舞伴配对问题：在舞会上，男、女各自排成一队。舞会开始时，依次从男队和女队的队头各出一人配成舞伴。如果两队初始人数不等，则较长的那一队中未配对者等待下一轮舞曲。假设初始男、女人数及性别已经固定，舞会的轮数从键盘输入。试模拟解决上述舞伴配对问题。要求：从屏幕输出每一轮舞伴配对名单，如果在该轮有未配对的，能够从屏幕显示下一轮第一个出场的未配对者的姓名。**

#include <malloc.h>

#include <stdio.h>

#define MAXSIZE 10

typedef struct

{

int \*base;

int front,rear;

}Man;

typedef struct

{

char \*base;

int front,rear;

}Woman;

void ManQueue(Man &Q,int m)//建立一个男人队列

{

Q.base =(int \*)malloc(sizeof(int)\*MAXSIZE);

if(!Q.base )return;

Q.front =Q.rear =0;

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

{

printf("输入第%d个男人名:",i+1);

scanf("%c",&Q.base[Q.rear]);

Q.rear =(Q.rear +1)% MAXSIZE;

getchar();

}

printf("输出男人队列：");

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

printf("%c ",Q.base[i]);

printf("\n");

}

void WomanQueue(Woman &Q,int n)//建立一个女人队列

{

Q.base =(char \*)malloc(sizeof(char)\*MAXSIZE);

if(!Q.base )return;

Q.front =Q.rear =0;

getchar();

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

{

printf("输入第%d个女人名:",i+1);

scanf("%c",&Q.base[Q.rear]);

Q.rear =(Q.rear +1)% MAXSIZE;

getchar();

}

printf("输出女人队列：");

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

printf("%c ",Q.base[i]);

printf("\n");

}

void peidui(Man Q1,Woman Q2,int m,int n,int k)//配对——k表示第几轮

{

int min,Mname;

char Wname;

if(m>n) min=n;else min=m;

for(int i=1;i<=k;i++)//第几轮

{

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

{

Mname=Q1.base[Q1.front];

Wname=Q2.base[Q2.front];

Q1.front=(Q1.front+1)%MAXSIZE;

Q2.front=(Q2.front+1)%MAXSIZE;

printf("配对者：%c--%c\n",Mname,Wname);

Q1.base [Q1.rear] =Mname;

Q1.rear =(Q1.rear +1)%MAXSIZE;

Q2.base [Q2.rear] =Wname;

Q2.rear =(Q2.rear +1)%MAXSIZE;

}

if(m>n)//男人多

{

printf("未配对的第一个出来的是：%c\n",Q1.base[Q1.front]);

}

else//女人多

{

printf("未配对的第一个出来的是：%c\n",Q2.base[Q2.front]);

}

}

}

void main()

{

Man Q1;

Woman Q2;

ManQueue(Q1,4);

WomanQueue(Q2,6);

peidui(Q1,Q2,4,6,3);

}