# 4

na=int(input())

a=[]

for i in range(0,na):

x=int(input())

a.append(x)

nb=int(input())

b=[]

for i in range(0,nb):

x=int(input())

b.append(x)

i=0

j=0

while (b[j]<=a[i]):

a.insert(i,b[j])

i+=1

j+=1

while (i<na and j<nb):

while (j<nb-1 and b[j]>a[i]): i+=1

while (b[j]<=a[i]):

a.insert(i,b[j])

i+=1

j+=1

while (j<nb):

a.append(b[j])

j+=1

print (a)

# 8

XSXXXSSSXXSXXSXXSSSS

# 15

for (i=1;i<=n;i++)

{

scanf("%d",&x);

Linklist \*p;

p=(Linklist\*)malloc(sizeof(Linklist));

p->data=x;

p->adr=(long long)(last)^(long long)(p);

}

while(p != head){

printf(" %d", p->data);

prepre = pre;

pre = p;

p = (Link\*)((long long)pre->preXORnext ^ (long long)(prepre));

}

# 16

#include <stdio.h>

#include <malloc.h>

typedef struct node

{

int data;

struct node \*next;

}Linklist;

Linklist \*head,\*last;

int n;

int main()

{

head=(Linklist\*)malloc(sizeof(Linklist));

last=head;

scanf("%d",&n);

int i,x,ans=-1;

Linklist \*p;

for (i=1;i<=n;i++)

{

scanf("%d",&x);

p=(Linklist\*)malloc(sizeof(Linklist));

p->data=x;

last->next=p;

p->next=NULL;

last=p;

}

scanf("%d",&x);

for (p=head->next;p!=NULL;p=p->next)

{

ans++;

if (p->data==x)

{

printf("%d\n",ans);

return 0;

}

}

p=(Linklist\*)malloc(sizeof(Linklist));

p->data=x;

last->next=p;

p->next=NULL;

for (p=head->next;p!=NULL;p=p->next) printf("%d ",p->data);

return 0;

}

# 19

#include <stdio.h>

#include <malloc.h>

typedef struct node{

int data;

struct node \* next;

}Link;

Link\* add(Link \*h,int x)

{

Link \*p;

p=(Link\*)malloc(sizeof(Link));

p->data=x;

p->next=NULL;

h->next=p;

return p;

}

void split(Link \*h1,Link \*h2)

{

int index=0;

Link \*q,\*pre=h1,\*last=h2;

for (q=h1->next;q!=NULL;q=q->next)

{

index++;

if (index%2==0)

{

last->next=q;

pre->next=q->next;

q->next=NULL;

last=q;

q=pre;

}

else pre=q;

}

return;

}

void print(Link \*h)

{

Link \*p;

for (p=h->next;p!=NULL;p=p->next) printf("%d ",p->data);

printf("\n");

return;

}

int main()

{

Link \*head1,\*last,\*head2;

head1=(Link\*)malloc(sizeof(Link));

head2=(Link\*)malloc(sizeof(Link));

last=head1;

int n,x,i;

scanf("%d",&n);

for (i=1;i<=n;i++)

{

scanf("%d",&x);

last=add(last,x);

}

split(head1,head2);

print(head1);

print(head2);

return 0;

}

# 22

#include <stdio.h>

#include <string.h>

#include <malloc.h>

typedef struct node{

char data;

struct node \* next;

}Link;

Link\* add(Link \*h,char x)

{

Link \*p;

p=(Link\*)malloc(sizeof(Link));

p->data=x;

p->next=NULL;

h->next=p;

return p;

}

void print(Link \*h,int len)

{

Link \*p;

int l=len;

for (p=h->next;p!=NULL&&l>0;p=p->next,l--) printf("%c",p->data);

printf("\n");

return;

}

void substr(Link \*h,int m,int n)

{

int i=0;

Link \*p=h;

for (;i<m;i++) p=p->next;

print(p,n);

return;

}

int main()

{

Link \*head1,\*last;

head1=(Link\*)malloc(sizeof(Link));

last=head1;

int l,i,m,n;

char s[100];

scanf("%s",s);

l=strlen(s);

for (i=0;i<l;i++) last=add(last,s[i]);

scanf("%d%d",&m,&n);

substr(head1,m,n);

return 0;

}