第一题答案：

#include <stdio.h>

void main()

{

int x = 1 ,find = 1;

while (find)

{

if ( x % 2 == 1 && x % 3 == 2 && x % 5 == 4 && x % 6 == 5 && x % 7 == 0)

{

printf("x = %d \n",x);

find = 0 ;

}

x++;

}

}

第二题答案：

#include <stdio.h>

#include <stdlib.h>

void main()

{

int x50,x20,x10,x5;

for(x50=1;x50<=9;x50++)

{

for(x20=1;x20<=21;x20++)

{

for(x10=1;x10<=42;x10++)

{

for(x5=1;x5<=84;x5++)

{

if(x50\*50+x20\*20+x10\*10+x5\*5==500 && x50+x20+x10+x5==30 )

printf("%d %d %d %d\n",x50,x20,x10,x5);

}

}

}

}

}

第三题答案：

#include "stdio.h"

void main()

{

char a[100];

int i,j=0,b,c,sum=0;

for(i=0;i<10;i++)

{

scanf("%d",&a[i]);

if(a[i]%2!=0&&j==0)

{

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

b=i;

j++;

continue;

}

else if(a[i]%2!=0&&j==1)

{

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

c=i;

j++;

continue;

}

}

for(i=0;i<10;i++)

if(i>b&&i<c)

sum=sum+a[i];

printf("第一个奇数下标为%d\n",b);

printf("第二个奇数下标为%d\n",c);

printf("二个奇数之间数的和为%d\n",sum);

}

第四题答案：

#include <stdio.h>

#include <string.h>

void reverse\_string(char \*start, char \*end)

{

while(start < end)

{

int tmp = \*start;

\*start = \*end;

\*end = tmp;

start++;

end--;

}

}

char \* left\_remove(char \* a, int k)

{

int len = strlen(a)-1;

char \*ret = a;

reverse\_string(a, a+len); // a b c d e f g h i j --> j i h g f e d c b a

reverse\_string(a, a+len-k); // j i h g f e d c b a --> c d e f g h i j b a

reverse\_string(a+len-k+1, a+len); // c d e f g h i j b a --> c d e f g h i j a b

return ret;

}

int main()

{

char a[100];

int kpoi;

printf("请输入一行字符串：");

gets(a);

printf("字符串左旋k位：k=");

scanf("%d",&kpoi);

printf("%s\n", left\_remove(a, kpoi));

}

第五题答案：

#include<stdio.h>

#include<stdlib.h>

#include<malloc.h>

#define ERROR 0;

typedef struct LNode

{

int data;

struct LNode \*next;

}LNode,\*LinkList;

LinkList InitList(LinkList L)

{

LinkList node = NULL;

node = (LinkList)malloc(sizeof(LNode));

if(!node)

{

return ERROR;

}

node->next = NULL;

L = node; return L;

}

int ListLength(LinkList L)

{

LinkList p = NULL;

int count = 0; p = L;

while(p->next)

{count++;p = p->next; }

return count;

}

LinkList CreateList\_L(LinkList L,int n)

{

int i;

LinkList p = NULL;

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

{

p = (LinkList)malloc(sizeof(LNode));

scanf("%d",&p->data);

p->next = L->next;

L->next = p;

}

return L;

}

LinkList ListInsert(LinkList L,int i,int e)

{

int j = 0;

LinkList s = NULL,p = NULL; p = L;

while(p&&j<i-1)

{

p = p->next; j++;

}

if(!p||j>i-1)

{

printf("输入的位置不合法！！！\n");

return L;

}

s = (LinkList)malloc(sizeof(LNode));

s->data = e;

s->next = p->next;

p->next = s;

return L;

}

LinkList ListDelete(LinkList L,int i)

{

int j;

LinkList q = NULL,p = NULL;

p = L; j = 0;

while(p->next&&j<i-1)

{ p = p->next; j++; }

if(!(p->next)||j>i-1)

{

printf("输入的位置不合法！！！\n");

return L;

}

q = p->next;

p->next = q->next;

free(q);

return L;

}

int GetElem(LinkList L,int i)

{

int j,e;

LinkList p = NULL;

if(i<1||i>ListLength(L))

{

printf("输入的位置不合法！！！\n");

return 0;

}

p = L->next;

j = 1;

while(j<i)

{ p = p->next; j++; }

e = p->data;

printf("第%d位的数据元素为%d\n",i,e);

return e;

}

void menu()

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*目录\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("输出单链表中的各元素值1\n");

printf("在单链表中插入数据元素2\n");

printf("在单链表中删除数据元素3\n");

printf("取出单链表中的数据元素4\n");

printf(" 结束程序0\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

void main()

{

int n,m,i,e;

LinkList L = NULL,p = NULL; L = InitList(L);

printf("请输入元素个数：");

scanf("%d",&n);

printf("依次输入%d个数据元素：",n);

L = CreateList\_L(L,n);

do

{

printf("\n \n");

menu();

printf("请输入你的选择：");

scanf("%d",&m);

switch(m)

{

case 1:

printf("现在链表的元素为：");

p= L->next;

while(p!=NULL)

{

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

p = p->next;

}

printf("\n");

break;

case 2:

printf("依次输入插入位置和数据元素(空格隔开)：");

scanf("%d %d",&i,&e);

L = ListInsert(L,i,e);

break;

case 3:

printf("输入需要删除的元素的位置：");

scanf("%d",&i);

L =ListDelete(L,i);

break;

case 4:

printf("输入需要取出的元素的位置：");

scanf("%d",&i);

GetElem(L,i);

break;

case 0:

printf("已结束程序！！！\n");

break;

default:

printf("输入错误！！！\n");

}

}while(m!=0); }