

作业 13

**必做题**

**第一题**

源代码:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/*学生信息结构体定义*/
struct Student{
    long long ID;
    char name[28];
    char sex[7];
    long long birthday;
    float score;
};

/*求平均值函数*/
float ffAverage(struct Student List[10]) {
    char cRound=0;
    float fSum=0;
    for (;cRound<10;cRound++) fSum+=List[cRound].score;
    return fSum/10;
}

/*排序和显示函数*/
int ifArngShow(struct Student List[10]) {
    char cRef=0, cProcess=1;
    struct Student *psAccord[10];
    psAccord[0]=List;
    for(;cProcess<10;cProcess++)
        psAccord[cProcess]=List+cProcess;
    for(;cRef<9;cRef++) {
        for (cProcess=cRef+1;cProcess<10;cProcess++) {
            if((*(psAccord+cProcess))->score>*(psAccord+cRef)->score) {
                struct Student *pTemp=*(psAccord+cProcess);
                *(psAccord+cProcess)=*(psAccord+cRef);
                *(psAccord+cRef)=pTemp;
            }
        }
    }
    printf("按照成绩由高到低输出: \n");
    for (cRef=0;cRef<10;cRef++)

    printf("%lld\t%s\t%s\t%lld\t%.2f\n", (*(psAccord+cRef))->ID, (*(psAccord+cRef)->name), (*(psAccord+cRef)->sex), (*(psAccord+cRef))->birthday, (*(psAccord+cRef))->score);
}
```

```

))->name, (*(psAccord+cRef))->sex, (*(psAccord+cRef))->birthday, (*(psAccord+cRef)
)->score);
    return 0;
}

/*主函数和输入*/
int main() {
    int i=0;
    struct Student List[10];
    /*输入提示*/
    printf("请按照如下格式输入10名学生的信息: \n学号(10位整数) 姓名(西文, 姓
名之间一个空格) 性别(male或female) 出生日期(8位整数) 成绩\n各信息间以空格
隔开. \n");
    for (;i<10;i++) {
        char cFlag=0;
        printf("第%d位学生: \n", i+1);
        while (cFlag==0) {
            long long llID=0, llBirthday=0;
            char cFirstName[25], cLastName[25], cSex[20];
            float rScore=100.0;

            scanf("%lld %s %s %s %lld %f", &llID, cFirstName, cLastName, cSex, &llBirthday, &
rScore);

            if ((llID<(long long)1E10 && llID>0) &&
                (strcmp(cSex, "male")==0 || strcmp(cSex, "female")==0) &&
                (llBirthday>19200000 && llBirthday<20240000) &&
                (rScore>=0.0 && rScore<=100.0)) {
                cFlag=1;
                List[i].ID=llID;
                strcpy(List[i].name, strcat(strcat(cFirstName, "
"), cLastName));

                strcpy(List[i].sex, cSex);
                List[i].birthday=llBirthday;
                List[i].score=rScore;
            }
            else printf("格式错误, 请重新输入: ");
        }
    }
    printf("学生成绩的平均值是: %f. \n", ffAverage(List));
    ifArngShow(List);
    system("pause");
    return 0;
}

```

运行结果截图:

```

请按照如下格式输入10名学生的信息：
学号（10位整数） 姓名（西文，姓名之间一个空格） 性别（male或female） 出生日期（8位整数） 成绩
各信息间以空格隔开。
第1位学生：
2023010001 Xiao Alpha male 20040901 60.0
第2位学生：
2023010002 Xiao Beta female 20041002 73.7
第3位学生：
2023010003 Xiao Gamma male 20041110 70.2
第4位学生：
2023010004 Xiao Delta male 20041203 99.2
第5位学生：
2023010004 Xiao Epsilon female 20050103 99.8
第6位学生：
2023010006 Xiao Zeta aaa 20050202 90.1
格式错误，请重新输入：2023010006 Xiao Zeta male 20050202 90.1
第7位学生：
2023010007 Xiao Ita female 20050307 92.5
第8位学生：
2023010008 Xiao Xi male 20050402 100.0
第9位学生：
2023010009 Xiao Miu female 20050401 10.2
第10位学生：
2023010010 Xiao Kappa male 20050506 85.3
学生成绩的平均值是：78.099998。
按照成绩由高到低输出：
2023010008 Xiao Xi male 20050402 100.00
2023010004 Xiao Epsilon female 20050103 99.80
2023010004 Xiao Delta male 20041203 99.20
2023010007 Xiao Ita female 20050307 92.50
2023010006 Xiao Zeta male 20050202 90.10
2023010010 Xiao Kappa male 20050506 85.30
2023010002 Xiao Beta female 20041002 73.70
2023010003 Xiao Gamma male 20041110 70.20
2023010001 Xiao Alpha male 20040901 60.00
2023010009 Xiao Miu female 20050401 10.20
请按任意键继续. . .

```

## 第二题

源代码：

```

#include <stdio.h>
#include <stdlib.h>
int main() {
    enum weekday {Monday=1, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday};
    enum weekday temp;
    scanf("%d",&temp);
    switch(temp) {
        case Monday:printf("Monday\n");break;
        case Tuesday:printf("Tuesday\n");break;
        case Wednesday:printf("Wednesday\n");break;
        case Thursday:printf("Thursday\n");break;
        case Friday:printf("Friday\n");break;
        case Saturday:printf("Saturday\n");break;
        case Sunday:printf("Sunday\n");break;
        default:printf("Wrong Input\n");
    }
    system("pause");
    return 0;
}

```

运行结果截图：

```
4
Thursday
请按任意键继续. . .
```

### 第三题

源代码:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define N 10

/*简化类型书写长度*/
typedef __int64 LI;
typedef unsigned __int64 LU;

/*分式结构体定义*/
typedef struct{
    LI numerator;
    LU denominator;
}fraction;

/*交换两数函数*/
int iSwap(LU *a, LU *b) {
    if (*a<*b) {
        LU temp=*a;
        *a=*b;
        *b=temp;
        return 1;
    }
    else return 0;
}

/*辗转相除约分函数*/
fraction Fra_Sim(fraction f) {
    LU a=_abs64(f.numerator);
    LU b=f.denominator;
    if (a==0) {
        f.denominator=1;
        return f;
    }
    iSwap(&a, &b);
    while ((a%b)!=0) {
        a-=b;
        iSwap(&a, &b);
    }
}
```

```

    }
    f.numerator/=(LI)b;
    f.denominator/=(LI)b;
    return f;
}

/*加法函数*/
fraction Fra_Add(fraction f1, fraction f2) {
    fraction sResult;
    /*检查输入合法性:
    这是题目要求，其实输出结果不会受有没有这个判断的影响。
    因为0*0=0, 0+0=0，非法输入正常计算结果就是0/0。*/
    if (f1.denominator==0 || f2.denominator==0)
        sResult.denominator=sResult.numerator=0;
    else{
        sResult.denominator=f1.denominator*f2.denominator;

        sResult.numerator=f1.numerator*f2.denominator+f1.denominator*f2.numerator;
    }
    return Fra_Sim(sResult);
}

/*减法函数*/
fraction Fra_Sub(fraction f1, fraction f2) {
    fraction sResult;
    /*检查输入合法性:
    这是题目要求，其实输出结果不会受有没有这个判断的影响。
    因为0*0=0, 00=0，非法输入正常计算结果就是0/0。*/
    if (f1.denominator==0 || f2.denominator==0)
        sResult.denominator=sResult.numerator=0;
    else{
        sResult.denominator=f1.denominator*f2.denominator;
        sResult.numerator=f1.numerator*f2.denominator-
f1.denominator*f2.numerator;
    }
    return Fra_Sim(sResult);
}

/*主函数-计算函数*/
int main() {
    fraction sum={0,1};
    char cRound=0;
    for (;cRound<(N/2);cRound++) {
        fraction add={1,4*cRound+1};

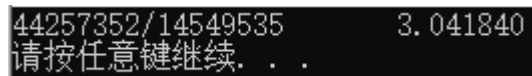
```

```

        fraction sub={1, 4*cRound+3};
        sum=Fra_Add(sum, add);
        sum=Fra_Sub(sum, sub);
    }
    if ((N%2)==1) {
        fraction add={1, 2*N-1};
        sum=Fra_Add(sum, add);
    }
    printf("%lld/%llu\t%lf\n", 4*sum.numerator, sum.denominator, 4.0*(double) sum.n
umerator/(double) sum.denominator);
    system("pause");
    return 0;
}

```

运行结果截图:



```

44257352/14549535      3.041840
请按任意键继续. . .

```

### 选做题

#### 第二题

源代码:

```

#include <stdio.h>
#include <stdlib.h>
/*定义链表相关结构体*/
#define LEN sizeof(Node)
typedef struct Node{
    unsigned int num;
    struct Node *index;
}Node;
char cFlag='0';

/*链表延伸*/
Node *pAdd(Node *pPrev){
    Node *pResult;
    unsigned int iInput=0;
    /*获取整数*/
    if (cFlag=='\n') return NULL;
    else{
        char cInput;
        while ((cInput=getchar())!='\n' && cInput!=' '){
            if (iInput!=0) iInput*=10;
            iInput+=(unsigned int)(cInput-'0');
        }
        (cInput=='\n')? (cFlag='\n'):(cFlag=' ');
    }
}

```

```

    }
    if ((pResult=(Node*)malloc(LEN))==NULL) {
        printf("内存超限。 \n");
        cFlag=0;
    }
    else{
        (*pPrev).index=pResult;
        (*pResult).num=iInput;
        (*pResult).index=NULL;
    }
    if (cFlag!=0) pAdd(pResult);
    return NULL;
}

/*删除节点*/
char del(Node *pInit) {
    Node *pLeastPrev=pInit;
    Node *pTemp;
    if ((*pInit).index==NULL) {
        free(pInit);
        return -52;
    }
    while ((*pInit=(*pInit).index).index!=NULL) {
        if ((*pInit).index).num<((*pLeastPrev).index).num)
            pLeastPrev=pInit;
    }
    printf("剩余结点中的最小结点值: %d\n", ((*pLeastPrev).index).num);
    pTemp=((*pLeastPrev).index).index;
    free((*pLeastPrev).index);
    (*pLeastPrev).index=pTemp;
    return 1;
}

/*主函数*/
int main() {
    /*创建列表*/
    Node *pInit;
    if ((pInit=(Node*)malloc(LEN))==NULL)
        printf("内存超限。 \n");
    else{
        (*pInit).num=0;
        pAdd(pInit);/*第一个结点没有保存数据*/
        /*录入失败释放空间*/
        if (cFlag==0) {
            Node *pNext=pInit;
            Node *pFree;

```

```

        while (pNext!=NULL) {
            pFree=pNext;
            pNext=(*pNext).index;
            free(pFree);
        }
    }
    else{
        while (del(pInit)!=-52) continue;
    }
}
system("pause");
return 0;
}

```

运行结果截图：

```

14654 143 72 277 390 12325 93376 123 10 331 17103
剩余结点中的最小结点值：10
剩余结点中的最小结点值：72
剩余结点中的最小结点值：123
剩余结点中的最小结点值：143
剩余结点中的最小结点值：277
剩余结点中的最小结点值：331
剩余结点中的最小结点值：390
剩余结点中的最小结点值：12325
剩余结点中的最小结点值：14654
剩余结点中的最小结点值：17103
剩余结点中的最小结点值：93376
请按任意键继续. . . |

```

### 第三题

huang, f, 81, 92, 173

### 第四题

源代码：

```

#include <stdio.h>
#include <stdlib.h>
#define LEN sizeof(Jew)
#define OVER 131072
#define CONTINUE 262144
/*定义链表元素*/
typedef struct Jew{
    char cIndex;
    struct Jew *pNext;
    struct Jew *pPrevious;
}Jew;
/*定义报数传递信息*/

```



```

typedef struct{
    char cFlag;
    Jew *p;
}Count;
Jew *pNow; /*正在报数的人*/
char cLen=41; /*还活着的人*/
char cOut; /*通用循环变量*/
/*建立链表*/
Jew *pSet(char cAmount){
    Jew *pInit, *pEnd, *pNew;
    if ((pInit=(Jew*)malloc(LEN))!=NULL) /*成功创建第一个结点*/
        pEnd=pInit;
        (*pInit).cIndex=0;
    for (cOut=1; cOut<cAmount; cOut++) /*创建后续结点*/
        if ((pNew=(Jew*)malloc(LEN))!=NULL) /*创建成功*/
            (*pNew).pPrevious=pEnd; /*本结点先前指针*/
            (*pEnd).pNext=pNew; /*前一节点后续指针*/
            pEnd=pNew; /*末结点后移*/
            (*pNew).cIndex=cOut; /*本结点序号*/
        }
    else /*创建失败，删除链表所有结点*/
        while (pInit!=pEnd){
            Jew* pRelease=pInit;
            pInit=(*pInit).pNext;
            free(pRelease);
        }
        free(pEnd);
        return NULL;
    }
}
/*创建循环关系*/
(*pNew).pNext=pInit;
(*pInit).pPrevious=pNew;
return pInit;
}
else return NULL;
}
/*删除结点*/
int iDel(Jew *pDel, char cAmount, char cLeft){
    if (cLen==++cLeft) /*完成筛选了*/
        printf("最终安全的位置是%d和%d号！\n", (*pDel).cIndex, ((*pDel).pNext).cIndex);
    /*释放空间*/
    free((*pDel).pNext);
}

```

```

        free(pDel);
        return OVER;
    }
    else{
        cLen--;
        printf("第%d轮出局者是%d号\n", cAmount-cLen, (*pDel).cIndex);
        /*删除结点*/
        ((*pDel).pPrevious).pNext=(*pDel).pNext;
        ((*pDel).pNext).pPrevious=(*pDel).pPrevious;
        free(pDel);
        return CONTINUE;
    }
}

/*单次报数*/
Count ccount(Count last){
    if (last.cFlag!=3){
        if (last.cFlag==2){
            Jew *pNext=(*last.p).pNext;
            if (iDel(last.p, 41, 2)==OVER) last.cFlag=3;
            else{
                last.cFlag=0;
                last.p=pNext;
            }
        }
        else{
            last.cFlag++;
            last.p=(*last.p).pNext;
        }
        ccount(last);
    }
    return last;
}

/*主函数*/
int main(){
    Jew *Init=pSet(41);
    if (Init==NULL) printf("电脑内存不够。");
    else{
        Count begin={0, Init};
        ccount(begin);
    }
    system("pause");
    return 0;
}

```

运行结果截图：

第1轮出局者是2号  
第2轮出局者是5号  
第3轮出局者是8号  
第4轮出局者是11号  
第5轮出局者是14号  
第6轮出局者是17号  
第7轮出局者是20号  
第8轮出局者是23号  
第9轮出局者是26号  
第10轮出局者是29号  
第11轮出局者是32号  
第12轮出局者是35号  
第13轮出局者是38号  
第14轮出局者是0号  
第15轮出局者是4号  
第16轮出局者是9号  
第17轮出局者是13号  
第18轮出局者是18号  
第19轮出局者是22号  
第20轮出局者是27号  
第21轮出局者是31号  
第22轮出局者是36号  
第23轮出局者是40号  
第24轮出局者是6号  
第25轮出局者是12号  
第26轮出局者是19号  
第27轮出局者是25号  
第28轮出局者是33号  
第29轮出局者是39号  
第30轮出局者是7号  
第31轮出局者是16号  
第32轮出局者是28号  
第33轮出局者是37号  
第34轮出局者是10号  
第35轮出局者是24号  
第36轮出局者是1号  
第37轮出局者是21号  
第38轮出局者是3号  
最终安全的位置是34和15号！  
请按任意键继续．．．|