

ASC C语言考核题解

第一题 牙牙学语

这题只是看看大家有没有做之前的50题，如果没有做也没事，有学习能力，在csdn这类读入有很多，很ez的。

```
#include <stdio.h>

int main()
{
    int temp = 0;
    while(scanf("%d", &temp) != EOF)
    {
        printf("%d\n", temp);
    }
    return 0;
}
```

第二题 饥肠辘辘

这题想考察大家基础语句的使用，会不会比较，会不会加减.....这些都是基本要求。但看到大家有些在这道题出问题，但基本都是输出格式问题，就不多赘述了。

```
#include <stdio.h>

int main()
{
    int T = 0, N = 0;
    int max_impression=0,temp_impression=0;
    int count =0,i;
    float v = 0.0,min_time = 0.0,temp_distance=0.0;
    int best = 0,near = 0;

    scanf("%d", &T);
    count = T;
    while (count-->0)
    {
        scanf("%d%f",&N,&v);

        max_impression = 0;
        min_time = 0.0;

        for(i=0;i<N;i++)
        {
            scanf("%f",&temp_distance);
            if(i==0 || temp_distance/v < min_time)
            {
                min_time = temp_distance/v;
                near = i+1;
            }
        }
    }
}
```

```

        for(i=0;i<N;i++)
        {
            scanf("%d",&temp_impression);
            if(temp_impression>max_impression)
            {
                max_impression = temp_impression;
                best = i+1;
            }
        }

        printf("%d is the best. The nearest canteen is %d. It takes
%.2fs.\n",best,near,min_time);
    }
}

```

第三题 苦心钻研

这题考察的就是基础的字符串读取的问题，大家基本没什么问题。但是这个PWM的原理，需要大家能够记住，之后单片机学习中会接触到。

```

#include <stdio.h>

int main()
{
    int N = 0, low = 0, high = 0;
    int count = 0,i = 0;
    char pwm[1000] = {'0'};
    float result;

    scanf("%d",&N);

    count = N;

    while(count--)
    {
        low = 0;high = 0;i=0;
        // getchar();
        scanf("%s", pwm);
        while(pwm[i] != '\0')
        {
            if(pwm[i] == '1')
            {
                high ++;
            }
            else if(pwm[i] == '0')
            {
                low ++;
            }
            i++;
        }

        result = high/(float)(low+high)*100;
    }
}

```

```
        printf("%.2f%%\n",result);
    }

    return 0;
}
```

第四题 初窥门径

这题考察的是大家二维数组的掌握情况，但是大家出现错误的点基本都是字符串上的问题，说明大家在学习字符串的时候没有学到位，也希望大家经过这一题，将字符串相关的知识复习一下。

```
#include <stdio.h>

int main()
{
    int T = 0, count_O = 0, count_X = 0;
    char board[3][4] = {'\0'};
    int count = 0, i, j;

    scanf("%d", &T);
    count = T;

    while(count--)
    {
        count_O = 0; count_X = 0;
        for(i=0; i<3; i++)
        {
            scanf("%s", board[i]);
        }
        for(i=0; i<3; i++)
        {
            for(j=0; j<4; j++)
            {
                switch (board[i][j])
                {
                    case 'O':
                        count_O ++;
                        break;
                    case 'X':
                        count_X ++;
                        break;
                    default:
                        break;
                }
            }
        }

        printf("There are %d loops and %d crosses.\n", count_O, count_X);
    }

    return 0;
}
```

```
}
```

第五题 判断胜负

这题的问题大多是我刚开始出题的时候，想着用0代表O后面又想到了O。导致出现了一些问题，所以责任在我。有些同学提出了一些复杂的情况（eg：只剩下一个棋子是否要判断下一步谁下，再继续判读胜负。但这个情况太复杂了，不适合放在这里考察大家的C语言），如果大家有兴趣，可以继续研究与探讨。

这题中题解运用了枚举的数据结构，也有用到typedef关键词的技巧，大家可以了解相关内容，这些知识大家可以进一步学习，在之后写大工程的时候增加代码可读性，可拓展性。

```
#include <stdio.h>

#define ROW 3
#define COL 3

typedef enum stage{
    Draw,
    O_win,
    X_win,
    Unfinished
}stage;

typedef enum role{
    player_none,
    player_O,
    player_X,
}role;

int is_full(char board[ROW][COL+1]);
role is_link(char board[ROW][COL+1]);
stage pd_win(char board[ROW][COL+1]);

int main()
{
    int T = 0;
    stage result = Draw;
    char board[ROW][COL+1] = {'\0'};
    int count = 0,i;

    scanf("%d",&T);
    count = T;

    while(count-->0)
    {
        for(i=0;i<ROW;i++)
        {
            scanf("%s",board[i]);
        }

        result = pd_win(board);
    }
}
```

```

        switch (result)
        {
        case Draw:
            printf("Draw\n");
            break;
        case O_win:
            printf("O_win\n");
            break;
        case X_win:
            printf("X_win\n");
            break;
        case Unfinished:
            printf("Unfinished\n");
            break;
        default:
            break;
        }
    }

    return 0;
}

int is_full(char board[ROW][COL+1])
{
    int i=0,j=0;

    for(i=0;i<ROW;i++)
    {
        for(j=0;j<COL;j++)
        {
            if(board[i][j]=='*')
            {
                return 0;
            }
        }
    }

    return 1;
}

role is_link(char board[ROW][COL+1])
{
    int i=0;
    char temp = '*';

    for(i=0;i<ROW;i++)
    {
        if(board[i][0] == board[i][1] && board[i][2] == board[i][1])
        {
            temp = board[i][0];
        }
    }
    for(i=0;i<COL;i++)
    {

```

```

        if(board[0][i] == board[1][i] && board[2][i] == board[1][i])
        {
            temp = board[0][i];
        }
    }
    if(board[0][0] == board[1][1] && board[2][2] == board[1][1])
    {
        temp = board[0][0];
    }
    if(board[0][2] == board[1][1] && board[2][0] == board[1][1])
    {
        temp = board[1][1];
    }

    switch (temp)
    {
        case '*':
            return player_none;
        case 'O':
            return player_o;
        case 'X':
            return player_x;
    }

    return player_none;
}

stage pd_win(char board[ROW][COL+1])
{
    int full = 0;
    role win = player_none;

    full = is_full(board);
    win = is_link(board);

    switch (win)
    {
        case player_none:
        {
            if(full)
            {
                return Draw;
            }
            else
            {
                return Unfinished;
            }
        }break;
        case player_o:
        {
            return o_win;
        }break;
        case player_x:
        {
            return x_win;
        }
    }
}

```

```

    }break;
    default:
        break;
    }
    return Draw;
}

```

第六题 临危受命

这道题大家遇到的问题很多，由于刚开始题目的表达问题，使得大家卡了好久，然后测试点也有些问题，也是在后续帮同学改代码的时候才发现。主要是思路的问题，导致大家的输出跟我想的不太一样。看题解可以看出来，我希望大家运用结构体进行存储信息，其中还有结构体套结构体的思路，然后发现没有合适的函数来进行日期比较，就必须构造函数来进行比较。这是一个比较合理的思路。

但是因为大家可能关注到了国庆返校然后就以为日期只能十月几号，而不是关注这个日期，所以导致思路不同。

虽然后期跟大家解释了，但是还是没人get到我点TT，不过也看到比较不错的代码。可以在给大家观赏一下。

这道题内核是排序，结构体的排序就有两种，一是如题解一样创建一个索引数组，二是使用指针，这里不做要求。有同学有使用值得表扬。

```

#include <stdio.h>
#include <string.h>

typedef struct date
{
    int month;
    int day;
}date;

typedef struct student
{
    char name[21];
    int dormitory;
    date departure;
    date back;
}student, *Pstudent;

void sort_1(student stu[], int index[], int n);
void sort_2(student stu[], int index[], int n);
void sort_3(student stu[], int index[], int n);
int datecmp(date date1, date date2);

int main()
{
    int T = 0, n=0, index[51] = {0};
    student stu[51];
    int count = 0,i;

    scanf("%d",&T);

    while(count<T)

```

```

{
    count ++;

    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%s %d %d.%d %d.%d",
            stu[i].name, &stu[i].dormitory,
            &stu[i].departure.month, &stu[i].departure.day,
            &stu[i].back.month, &stu[i].back.day);
        index[i] = i;
    }

    sort_1(stu,index,n);
    printf("Dormitory %d:\n",count);
    for(i=0;i<n;i++)
    {
        printf("%s %d %d.%d %d.%d\n",
            stu[index[i]].name, stu[index[i]].dormitory,
            stu[index[i]].departure.month, stu[index[i]].departure.day,
            stu[index[i]].back.month, stu[index[i]].back.day);
    }

    sort_2(stu, index, n);
    printf("Departure %d:\n",count);
    for(i=0;i<n;i++)
    {
        printf("%s %d %d.%d %d.%d\n",
            stu[index[i]].name, stu[index[i]].dormitory,
            stu[index[i]].departure.month, stu[index[i]].departure.day,
            stu[index[i]].back.month, stu[index[i]].back.day);
    }

    sort_3(stu, index, n);
    printf("Back %d:\n",count);
    for(i=0;i<n;i++)
    {
        printf("%s %d %d.%d %d.%d\n",
            stu[index[i]].name, stu[index[i]].dormitory,
            stu[index[i]].departure.month, stu[index[i]].departure.day,
            stu[index[i]].back.month, stu[index[i]].back.day);
    }
}

return 0;
}

int datecmp(date date1, date date2)
{
    if(date1.month > date2.month)
    {
        return 1;
    }
    else if(date1.month < date2.month)
    {
        return -1;
    }
}

```



```

    }
    else
    {
        if(date1.day == date2.day)
        {
            return 0;
        }
        else if(date1.day>date2.day)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }
}

void sort_1(student stu[], int index[], int n)
{
    int i,j,count = 0;
    int temp;

    for(i=0;i<n-1;i++)
    {
        count = 0;
        for(j=0;j<n-1-i;j++)
        {
            if(stu[index[j]].dormitory>stu[index[j+1]].dormitory ||
                (strcmp(stu[index[j]].name,stu[index[j+1]].name)>0 &&
stu[index[j]].dormitory == stu[index[j+1]].dormitory))
            {
                temp = index[j];
                index[j] = index[j+1];
                index[j+1] = temp;
                count = 1;
            }
        }
        if(count == 0)
            break;
    }
}

void sort_2(student stu[], int index[], int n)
{
    int i,j,count = 0;
    int temp;

    for(i=0;i<n-1;i++)
    {
        count = 0;
        for(j=0;j<n-i-1;j++)
        {
            if(datecmp(stu[index[j]].departure,stu[index[j+1]].departure)>0 ||

```

```

        (datecmp(stu[index[j]].departure,stu[index[j+1]].departure)==0
        && strcmp(stu[index[j]].name,stu[index[j+1]].name)>0))
    {
        temp = index[j];
        index[j] = index[j+1];
        index[j+1] = temp;
        count = 1;
    }
}
if(count == 0)
    break;
}
}

void sort_3(student stu[], int index[], int n)
{
    int i,j,count = 0;
    int temp;

    for(i=0;i<n-1;i++)
    {
        count = 0;
        for(j=0;j<n-i-1;j++)
        {
            if(datecmp(stu[index[j]].back,stu[index[j+1]].back)>0 ||
                (datecmp(stu[index[j]].back,stu[index[j+1]].back)==0 &&
                strcmp(stu[index[j]].name,stu[index[j+1]].name)>0))
            {
                temp = index[j];
                index[j] = index[j+1];
                index[j+1] = temp;
                count = 1;
            }
        }
        if(count == 0)
            break;
    }
}

```

第七题 帮助队友

菜单这题其实是没什么难度的。主要是如何去构建一个菜单。一般这种简单的菜单用一个stage变量和一个cursor变量再外加一堆判断就能实现。

但其实菜单还可以通过结构体来实现，我很高兴，发现有同学使用了结构体。还有代码规范比较好的。也希望大家多多学习交流。

```

#include <stdio.h>

void Menu_Show(void);

```

```

int stage=0,cursor=0;

int main()
{
    int operate;
    scanf("%d",&operate);
    Menu_Show();
    while(operate!=0)
    {

        switch (operate)
        {
            case 1:
            {
                switch (stage)
                {
                    case 0:
                        cursor = (cursor+1)%2;
                        break;
                    case 1:
                        cursor = (cursor+2)%3;
                        break;
                }
            }break;
            case 2:
            {
                switch (stage)
                {
                    case 0:
                        cursor = (cursor+1)%2;
                        break;
                    case 1:
                        cursor = (cursor+1)%3;
                        break;
                }
            }break;
            case 3:
            {
                switch (stage)
                {
                    case 0:
                    {
                        switch (cursor)
                        {
                            case 0:
                            {
                                cursor = 0;
                                stage = 1;
                            }break;
                            case 1:
                            {
                                stage = 2;
                                cursor = 0;
                            }break;
                        }
                    }
                }
            }
        }
    }
}

```

```

        }break;
    }break;
    case 4:
    {
        switch (stage)
        {
            case 1:
            case 2:
            {
                stage = 0;
                cursor = 0;
            }break;
        }
    }break;
    default:
        break;
    }
}
Menu_Show();
scanf("%d",&operate);
}
return 0;
}

```

```

void Menu_Show()
{
    switch (stage)
    {
        case 0:
        {
            printf("Main\n");
            if(cursor == 0)
            {
                printf("->  data\n    image\n");
            }
            else if(cursor == 1)
            {
                printf("    data\n->  image\n");
            }
        }break;
        case 1:
        {
            printf("data\n");
            switch (cursor)
            {
                case 0:
                {
                    printf("->  data1 101\n    data2 20.1\n    data3 1.23\n");
                    break;
                }
                case 1:
                {
                    printf("    data1 101\n->  data2 20.1\n    data3 1.23\n");
                    break;
                }
                case 2:
                {
                    printf("    data1 101\n    data2 20.1\n->  data3 1.23\n");
                    break;
                }
                default:
            }
        }
    }
}

```

```
        break;
    }
}break;
case 2:
{
    printf("image\n    ASC\n");
}break;

default:
    break;
}
}
```

寄语

这次的c语言考察在一系列的问题中也是差不多结束了，但是大家学习智能车并没有结束。智能车的学习是漫长，希望大家能够坚持不懈的坚持下去。之后会陆续发布各个模块的任务，希望大家能够继续像这几天这样积极学习。当然也要劳逸结合。