

作业 10

必做题

第一题

源代码:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char cfCmp(char sA[31], char sB[31], char cLoc) { /*判断该位是否匹配*/
    char cCmpRound=0, cCmpFlag=1; /*循环位置变量和标志变量 (0为不匹配, 1为匹配)
    */
    char c=strlen(sB);
    for (; cCmpRound<strlen(sB); cCmpRound++) {
        char cA=sA[cCmpRound], cB=sB[cCmpRound];
        if ((sA[cCmpRound+cLoc]!=sB[cCmpRound]) && (sB[cCmpRound]!='?')) { /*匹配
        失败*/
            cCmpFlag=0;
            break;
        }
    }
    if (cCmpFlag==1) return cLoc; /*匹配成功*/
    else return -52; /*匹配失败*/
}

int main() { /*主函数*/
    char sA[31], sB[31], cRound=0, cFlag=0;
    /*cRound为循环位置变量; cFlag为可否匹配的标志, 1可, 0不可*/
    /*输入sA*/
    printf("sA=");
    scanf("%s", sA);
    /*输入sB*/
    printf("sB=");
    scanf("%s", sB);
    /*找到并输出每一个匹配位置*/
    for (; cRound<(strlen(sA)+1-strlen(sB)); cRound++) {
        char cResult=cfCmp(sA, sB, cRound);
        if (cResult!=-52) { /*找到匹配*/
            printf("%hd\t", cResult);
            cFlag=1;
        }
    }
    /*输出收尾*/
    if (cFlag) printf("are all the matches found.\n");
    else printf("No match found.\n");
    system("pause");
    return 0;
}
```

```
}
```

运行结果:

```
sA=abcdefghc*exyzcferk
sB=c?e
2      8      14      are all the matches found.
请按任意键继续. . . |
```

第二题

源代码:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/*数组定义*/
char cWord[28];/*单词储存*/
char cPoint[28];/*元音位置数组*/
char cVowel[28];/*元音字母数组*/
/*找元音程序*/
char cfFindV(void) {
    char *pPoint=cPoint, *pVowel=cVowel, *pWord=cWord;/*指向数组进行到的位置*/
    for (;*(pWord)!='\0';pWord++) {
        switch (*(pWord)) {/*是元音*/
            case 'a':
            case 'e':
            case 'i':
            case 'o':
            case 'u': {
                *(pVowel)=*(pWord);/*该字母添加入元音字母数组*/
                *(pPoint)=(char) (pWord-cWord);/*该位置添加入元音位置数组*/
                /*指标增加*/
                pVowel++;
                pPoint++;
                break;
            }
        }
    }
    return 0;
}

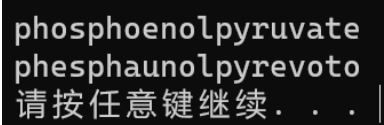
/*补回元音程序*/
char cfBackV(void) {
    char cRoundBack=0;
    for (;cPoint[cRoundBack]!='\0';cRoundBack++) {/*按有元音的位置遍历*/
        cWord[(int) cPoint[cRoundBack]]=cVowel[strlen(cPoint)-cRoundBack-1];
    }
}
```

```

        return 0;
    }
    /*主函数*/
    int main() {
        scanf("%s", cWord); /*输入*/
        cfFindV();
        cfBackV();
        puts(cWord); /*输出*/
        system("pause");
        return 0;
    }

```

运行结果：



```

phosphoenolpyruvate
phesphaunolpyrevoto
请按任意键继续. . .

```

第三题

源代码：

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char cWord[64]; /*给定字母组合*/
/*插入排序函数*/
char fcInsert(char cLoc) {
    if (cLoc > 0) {
        char cRound = cLoc, cTemp; /*循环变量和交换临时变量*/
        fcInsert(cLoc - 1); /*处理前面所有位的排序*/
        for (; cRound > 0; cRound--) {
            if (cWord[cRound] > cWord[cRound - 1]) { /*相邻两位置大小反了就交换位置*/
                cTemp = cWord[cRound];
                cWord[cRound] = cWord[cRound - 1];
                cWord[cRound - 1] = cTemp;
            }
        }
    }
    return 0;
}
/*主函数*/
int main() {
    scanf("%s", cWord); /*输入*/
    fcInsert(strlen(cWord) - 1); /*排序*/
    printf("%s\n", cWord); /*输出*/
    system("pause");
}

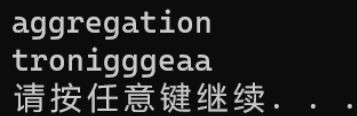
```

```

        return 0;
}

```

运行结果：



```

aggregation
tronigggeaa
请按任意键继续...

```

选做题

第一题

源代码：

```

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

static short sASCII1[128];/*第一个字符串的字符分布*/
static short sASCII2[128];/*第二个字符串的字符分布*/
static char cTemp[30];/*输入数组临时存放*/
/*存放字符分布的函数*/
char cfPutIn(short sASCII[128],char cTemp[30]){
    char cRound=0;/*循环变量*/
    char a;
    for (cRound=0;cTemp[cRound]!='\0';cRound++){/*把字符赋给字符分布中对应位置*/
        a=sASCII[cTemp[cRound]];
        sASCII[a]++;
    }
    return 0;
}

/*比较数组是否相同函数*/
char cfCmp(short sASCII1[128],short sASCII2[128]){
    short sRound=0; /*循环变量*/
    for (;sRound<128;sRound++){
        /*存在不同*/
        if (sASCII1[sRound]!=sASCII2[sRound]) return 0;
    }
    /*完全相同*/
    return 1;
}

/*主函数*/
int main() {
    /*输入并整理第一个单词*/
    printf("请输入第一个单词：");
    scanf("%s",cTemp);
}

```

```

    cfPutIn(sASCII1, cTemp);
    /*输入并整理第二个单词*/
    printf("请输入第二个单词: ");
    scanf("%s", cTemp);
    cfPutIn(sASCII2, cTemp);
    /*比较并输出结果*/
    printf("%hd", cfCmp(sASCII1, sASCII2));
    system("pause");
    return 0;
}

```

运行结果:

```

请输入第一个单词:alore
请输入第二个单词:lareo
1请按任意键继续. . . |

```

第二题

源代码:

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
/*可否转换相关宏*/
#define NOTOK 0
#define OK 1
/*储存十进制表示的地址*/
char cOrigin[13];
/*平移三位函数*/
char cMove(char cNextOrigin) {
    char cRoundMove=cNextOrigin*3;
    for (;cRoundMove<(cNextOrigin+1)*3;cRoundMove++)
        cOrigin[cRoundMove]=cOrigin[cRoundMove+1];
    return 0;
}
/*三位求和函数*/
short sSumThree(char cNextOrigin) {
    short sSum=0;
    char cRoundSum=0;
    for (;cRoundSum<3;cRoundSum++) {
        char cTemp=cOrigin[cRoundSum+cNextOrigin*3];
        if (cTemp!=0) { /*非空位求和*/
            sSum*=10;
            sSum+=cTemp-48;
        }
    }
}

```

```

    }
    return sSum;
}

/*三位录入补零函数*/
char cPut(const char str[],char cLocStr,char cNextOrigin){
    /*cLocStr为下一个要处理的字符串位,
    cNextOrigin为处理组数的序号*/
    char cLocOrigin=(cNextOrigin+1)*3, cFlag=0; /*填入cOrigin的初始位置*/
    while (str[cLocStr]!='.') {
        if (isdigit(str[cLocStr])==0 || cFlag==3) return -52; /*数据异常*/
        cOrigin[cLocOrigin]=str[cLocStr];
        cLocStr++;
        cMove(cNextOrigin);
        cFlag++;
    }
    return cLocStr;
}

/*转换函数*/
int aton(const char str[]) {
    char cInput[20], cLocStrMain=0, cNextOriginMain=0, cRoundAdd=0;
    int iSumMain=0;
    strcpy(cInput, str);
    cInput[strlen(str)]='.';
    /*整理每三位填入cOrigin*/
    while (cLocStrMain!=(strlen(str)+1)) {
        cLocStrMain=cPut(cInput,cLocStrMain,cNextOriginMain++);
        if (cLocStrMain== -52) return 0;
        cLocStrMain++;
    }
    /*对cOrigin每三位求和*/
    for (;cRoundAdd<4;cRoundAdd++) {
        iSumMain*=256;
        iSumMain+=sSumThree(cRoundAdd);
    }
    return iSumMain;
}

/*主运行测试函数*/
int main() {
    char cStr[20];
    scanf("%s", cStr);
    printf("%d\n", aton(cStr));
    system("pause");
    return 0;
}

```

运行结果:

```
166.111.14.89  
-1502671271  
请按任意键继续. . .
```