

安徽大学 20 22 —20 23 学年第 1 学期

《高级语言程序设计》考试试卷 (A 卷)

参考答案及评分标准

一、阅读程序题 (30 分, 1、2 小题各 5 分, 3、4 小题各 10 分)

1. 参考答案

x=3, y=2, z=6 (3 分) 除数字外格式正确 2 分, 数字 1 分
a=-40 (2 分) 除数字外格式正确 1 分, 数字 1 分

2. 参考答案

a=0 (2 分) 除数字外格式正确 1 分, 数字 1 分
a=2 (3 分) 除数字外格式正确 1 分, 数字 2 分

3. 参考答案

a[3][2]=9 (字母 a 2 分, 每个 [] 各 1 分, 每个数字 2 分)

4. 参考答案

1 15 21 175 305 除数字外格式正确 2 分, 再次基础上, 对 1 个给 4 分, 对 2 个给 6 分, 对 3 个给 8 分, 对 4 个给 9 分, 对 5 个给 10 分

二、程序改错题 (10 分, 每小题 5 分)

1. 参考答案

```
01 #include<stdio.h>_____
02 #define N 5_____
03 int main(){_____ void main() 或此处不改, main 结尾加 return 0; _____}
04 int a[N], i, temp;_____
05 printf("enter array a: \n");_____
06 for(i=0; i<N; i++)_____
07 scanf("%d", a[i]); scanf("%d", &a[i]); _____
08 printf("array a: \n");_____
09 for(i=0; i<N; i++)_____
10 printf("%4d", a[i]);_____
```

```

11  for(i=0; i<N; i++) for(i=0; i<N/2; i++)
12      {   temp = a[i];
13          a[i] = a[N-i]; a[i] = a[N-i-1];
14          a[N-i] = temp; a[N-i-1] = temp;
15      }
16  printf("\nNow, array a: \n");
17  for(i=0; i<N; i++)
18      printf("%4d", a[i]);
19  printf("\n");
20 }

```

评分标准：每处 1 分，其他改正方法只要正确酌情给分

2. 参考答案

```

01 #include<stdio.h>
02 int main(){
03     char *pSrc="Anhui University", tmp[100];
04     char *pDest; char *pDest=tmp;
05     while(*pDest!=0){ while(*pSrc!=0){ 或 while(*pSrc!='\0'){
06         (*pDest)++ = (*pSrc)++; *pDest++ = *pSrc++; 或 *(pDest++) = *(pSrc++);
07     }
08     *pDest='\0'; *pDest='\0'; 或 *pDest=0;
09     printf("%s", *pDest); printf("%s", tmp); 或 pDest=tmp; printf("%s", *pDest);
10     return 0;
11 }

```

评分标准：每处 1 分，其他改正方法只要正确酌情给分

三、分析填空题（20 分，每小题 10 分）

1. 参考答案（每空 2 分，共 10 分）：

- ① {&a, &b, &c, &d, &e}
- ② *p[i] 或 ** (p+i) 或 2*i+1 或 2* (i+1) -1 或 2*i+a
- ③ *p[i] 或 ** (p+i)
- ④ pa[i] 或 *(pa+i)
- ⑤ *pp

2. 参考答案（每空 2 分，共 10 分）

- ① &a[i] （对应 scanf 函数内的空格）
- ② int array[] 或 int array[10] 或 int *array
- ③ array[j] > array[k] （array[j] < array[k]也可酌情算正确）
- ④ array[k]=array[i];

⑤ array[i]=t;

只要答案正确，均可酌情给分；排序顺序可以酌情不予考虑。

四、程序设计题（40 分，每小题 20 分）

1. 参考答案

```
#include <stdio.h>
#include <math.h>
int main()
{
    int s;
    float n,t,pi;
    t=1; pi=0; n=1.0; s=1;
    while((fabs(t))>=1e-6)
    {
        pi=pi+t;
        n=n+2;
        s= -s;
        t=s/n;
    }
    pi=pi*4;
    printf("pi=%10.6fn",pi);
    return 0;
}
```

评分标准：

有预处理指令 2 分

变量定义与使用正确 2 分

循环语句内的条件表达式正确 2 分

能够正确用循环进行运输 4 分

输出语句书写正确 2 分

程序逻辑正确 4 分

程序结果正确 4 分

2. 参考答案

2.1 链表实现方式：

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define NUM 3
#define LENSTU sizeof(struct Student)
#define LENLAPTOP sizeof(struct Laptop)

typedef struct Laptop{ //6 分
    char ComType[20];
```

```

        char ComColor[10];
        float ComPrice;
    }LaptopNode;

typedef struct Student{    //6 分
    char StuNum[20];
    char StuName[20];
    char StuGender[20];
    char StuMajor[20];
    struct Laptop* StuLaotop;

    struct Student * next;
}StuNode;

struct Student * insert(StuNode *, StuNode *);
struct Student * del(StuNode *, char[]);
StuNode * query(StuNode *, char[]);
void print(StuNode *);

void main(){
    StuNode * pt=NULL;
    int i;
    for(i=0;i<NUM;++i){
        StuNode *pStu=(StuNode *) malloc(LENSTU);
        pStu->StuLaotop=(LaptopNode *) malloc(LENLAPTOP);
        printf("Please input the information of a new student, No.:%d:\n", i+1);
        scanf("%s", pStu->StuNum);
        scanf("%s", pStu->StuName);
        scanf("%s", pStu->StuGender);
        scanf("%s", pStu->StuMajor);
        scanf("%s", pStu->StuLaotop->ComType);
        scanf("%s", pStu->StuLaotop->ComColor);
        scanf("%f", &pStu->StuLaotop->ComPrice);
        pt=insert(pt,pStu);
        print(pt);
    }

    char StuNum[20];
    printf("Please input the num of the student that you want to query:\n");
    scanf("%s", StuNum);
    pt=query(pt, StuNum);

    char StuName[20];
    printf("Please input the name of the student that you want to delete:\n");
    scanf("%s", StuName);
    pt=del(pt, StuName);
    print(pt);
}

StuNode * insert(StuNode *pHead, StuNode *pStu){    //2 分
    StuNode *pFront,*pRear;
    pFront=pHead;
    if(pHead==NULL){
        pHead=pStu;
        pStu->next=NULL;
    }
    else{
        pRear=pHead;

```

```

while(pRear->next!=NULL){
    pFront=pRear;
    pRear=pRear->next;
}
pRear->next=pStu;
pStu->next=NULL;
}
printf("Student %s is inserted!\n", pStu->StuName);
return(pHead);
}

```

```

StuNode *del(StuNode *pHead, char StuName[20]){ //2 分
    StuNode *pFront, *pRear;
    if (pHead==NULL){
        printf("\nlist null!\n");
        return pHead;
    }

    pFront=pHead;
    pRear=pHead;
    while(0!=strcmp(StuName, pRear->StuName) && pRear->next!=NULL){
        pFront=pRear;
        pRear=pRear->next;
    }
    if(0==strcmp(StuName, pRear->StuName)){
        if(pRear==pHead)
            pHead=pRear->next;
        else
            pFront->next=pRear->next;
        printf("Student %s is deleted!\n", pRear->StuName);
    }
    free(pRear);
    return(pHead);
}

```

```

StuNode *query(StuNode *pHead, char StuNum[20]){ //2 分
    StuNode *pFront, *pRear;
    if (pHead==NULL){
        printf("\nlist null!\n");
        return pHead;
    }
    pFront=pHead;
    pRear=pHead;
    while(0!=strcmp(StuNum, pRear->StuNum) && pRear->next!=NULL){
        pFront=pRear;
        pRear=pRear->next;
    }
    if(0==strcmp(StuNum, pRear->StuNum)){
        printf("StuNum:%s, StuName:%s, StuGender:%s, StuMajor:%s, ComType:%s, ComColor:%s,
ComPrice: %f\n"
            ,pRear->StuNum, pRear->StuName, pRear->StuGender, pRear->StuMajor
            ,pRear->StuLaotop->ComType, pRear->StuLaotop->ComColor, pRear->StuLaotop->ComPrice);
    }
    else{
        printf("Student %s cannot be found!\n");
    }
    return(pHead);
}

```

```

void print(StuNode * pStu){           //2 分
    if (pStu==NULL){
        printf("\nlist null!\n");
        return;
    }
    while(pStu!=NULL){
        printf("StuNum:%s,  StuName:%s,  StuGender:%s,  StuMajor:%s,  ComType:%s,  ComColor:%s,
ComPrice: %f\n"
                ,pStu->StuNum, pStu->StuName, pStu->StuGender, pStu->StuMajor
                ,pStu->StuLaotop->ComType, pStu->StuLaotop->ComColor, pStu->StuLaotop->ComPrice);
        pStu=pStu->next;
    }
}

```

2.2 数组实现方式

```

#include <stdio.h>
#include<stdlib.h>
#include<string.h>
#define LENSTU sizeof(struct Student)
#define LENLAPTOP sizeof(struct Laptop)
#define NUM 3
#define ArrayMax 20

```

```

typedef struct Laptop{           //6 分
    char ComType[20];
    char ComColor[20];
    float ComPrice;
}LaptopNode;

```

```

typedef struct Student{         //6 分
    char StuNum[20];
    char StuName[20];
    char StuGender[20];
    char StuMajor[20];
    struct Laptop* StuLaotop;
}StuNode;

```

```

int _ArrayLen=0;
StuNode * pt[ArrayMax]={NULL};

```

```

void insert();
void del(char[]);
void query(char[]);
void print();

```

```

void main(){
    int i;
    for(i=0;i<NUM;++i){
        insert();
        print();
    }

    char StuNum[20];
    printf("Please input the num of the student that you want to query:\n");
    scanf("%s", StuNum);
    query(StuNum);
}

```

```

        char StuName[20];
        printf("Please input the name of the student that you want to delete:\n");
        scanf("%s", StuName);
        del(StuName);

        print();
    }

    void insert(){ //2 分
        StuNode *pStu=(StuNode *) malloc(LENSTU);
        pStu->StuLaotop=(LaptopNode *) malloc(LENLAPTOP);
        printf("Please input the information of a new student:\n");
        scanf("%s", pStu->StuNum);
        scanf("%s", pStu->StuName);
        scanf("%s", pStu->StuGender);
        scanf("%s", pStu->StuMajor);
        scanf("%s", pStu->StuLaotop->ComType);
        scanf("%s", pStu->StuLaotop->ComColor);
        scanf("%f", &pStu->StuLaotop->ComPrice);

        pt[_ArrayLen]=pStu;
        ++_ArrayLen;
        printf("Student %s is inserted!\n", pStu->StuName);
    }

    void del(char StuName[20]){ //2 分
        if (_ArrayLen==0){
            printf("\nlist null!\n");
            return;
        }
        int i=0;
        StuNode *pRear=NULL;
        for(i=0;i<_ArrayLen;++i){
            pRear=pt[i];
            if(0==strcmp(StuName, pRear->StuName)){
                int j=i;
                if(j==_ArrayLen-1){
                    pt[j]=NULL;
                }
                else{
                    for(;j<_ArrayLen-1;++j){
                        pt[j]=pt[j+1];
                    }
                }
                --_ArrayLen;

                printf("Student %s is deleted!\n", pRear->StuName);
                break;
            }
        }
    }

    void query(char StuNum[20]){ //2 分
        if (_ArrayLen==0){
            printf("\nlist null!\n");
            return;
        }
    }

```

```

int i=0, iflag=0;
StuNode *pRear=NULL;
for(i=0;i<_ArrayLen;++i){
    pRear=pt[i];
    if(0==strcmp(StuNum, pRear->StuNum)){
        printf("StuNum:%s, StuName:%s, StuGender:%s, StuMajor:%s, ComType:%s, ComColor:%s,
ComPrice: %f\n"
                ,pRear->StuNum, pRear->StuName, pRear->StuGender, pRear->StuMajor
                ,pRear->StuLaotop->ComType, pRear->StuLaotop->ComColor,
pRear->StuLaotop->ComPrice);

        iflag=1;
        break;
    }
}
if(iflag==0){
    printf("Student %s cannot be found!\n");
}
}

void print(){ //2 分
    if (_ArrayLen==0){
        printf("\nlist null!\n");
        return;
    }
    int i=0;
    StuNode *pRear=NULL;
    for(i=0;i<_ArrayLen;++i){
        pRear=pt[i];
        printf("StuNum:%s, StuName:%s, StuGender:%s, StuMajor:%s, ComType:%s, ComColor:%s,
ComPrice: %f\n"
                ,pRear->StuNum, pRear->StuName, pRear->StuGender, pRear->StuMajor
                ,pRear->StuLaotop->ComType, pRear->StuLaotop->ComColor, pRear->StuLaotop->ComPrice);
    }
}

```