

答案

1- 回车（光标跳到当前行首）

水平制表（HT）

\"

\\

2- 没有，短路与

3- 7

4- static 修饰外部变量，改变了它的作用域（文件作用域）；函数内的变量，改变了它的生存期

5- 字符串数组

指向整型变量的指针数组

6-否

是

不能改变原左值数组，数组名指针指向了其它位置；而通过函数，是创建了一个新的指针临时变量。

7- s(1+4=5 字节) u(max(4,8) = 8 字节)

8- 有，返回了指向内部临时数组的指针

9- 可以

struct 默认权限是public, class 是 private

10- 多态

第一道大题

```
1  int a[100] = {-1};
2
3  int cmp(const void *a,const void *b)
4  {
5      return (*(int *)b)-*(int *)a); //实现的是降序排序
6  }
7
8  long moreBigger(long in){
9      int index = 0;
10     do{
11         a[index] = in %10;
12         index++;
13     }while(in /= 10);
```

```

14
15     qsort(a,100,sizeof(a[0]),cmp);    /// <stdlib.h>
16     long res = 0;
17     for(int j =0; j < index; j++){
18         res = res*10 + a[j];
19     }
20
21     return res;
22 }

```

第二道大题

```

1  bool Judge(){
2      char str[100];
3      printf("Enter a message:");
4      //  gets(str);    ///  不能用 scanf
5      fgets(str, (sizeof str / sizeof str[0]), stdin);
6      char * p = strrchr(str, '\0');
7
8      printf("%d",p - str);
9      char newStr[100];
10     int p_newStr = 0;
11     for(int i=0; i < p - str; i++){
12         if( (str[i] >='a' && str[i] <= 'z' || str[i] >= 'A' && str[i] <=
13         'Z' ) ){
14             newStr[p_newStr] = str[i];
15             p_newStr++;
16         }
17     }
18     newStr[p_newStr] = '\0';
19     printf("%s", newStr);
20
21     char * pp = strrchr(newStr, '\0');
22     char * end = pp-1;
23     char * begin = newStr;
24     bool flag = true;
25     while (begin < end){
26         if( *begin != *end ){
27             flag = false;
28             break;
29         }else{
30             begin++;
31             end--;
32         }
33     }
34     return flag;
35

```

第三道大题

```
1 void find(){
2     int *array;
3     int N;
4     printf("请输入所要创建的一维动态数组的长度: ");
5     scanf("%d",&N);
6     array=(int*)calloc(N+1,sizeof(int));
7
8     for(int i=2; i <=N; i++){
9         array[i] = 0;
10    }
11
12    for(int i=2; i<=N; i++){
13        if(!array[i]){
14            for(int j = i; i*j <=N; j++)
15                array[i*j] = 1;
16        }
17    }
18
19    int j=0;
20    for(int i=2;i<=N;i++)
21    {
22        if(!array[i])
23        {
24            printf("%d ",i);
25            j++;
26            if(j%10 == 0)
27                printf("\n");
28        }
29    }
30    printf("\n");
31    free(array);
32 }
```

第四道题

```
1 int main() {
2     double precise;
3     scanf("%lf", &precise);
4     double res = 0;
5     int index = 1;
6     long long prod = 1;
7     if(precise > 0){
8         while(1){
```

```

9         prod = prod * index;
10        double current = 1.0 / prod;
11        if( current < precise ){
12            break;
13        }
14        res = res + current;
15        index++;
16    }
17    res = res + 1;
18    printf("%lf", res);
19 }else{
20     printf("Precision must be positive float");
21 }
22
23 return 0;
24 }

```

第五道题

```

1  typedef struct NODE
2  {
3      char* val;
4      struct NODE *next;
5  }node; //define node
6
7  node * head=NULL; //init head_node of stack
8  int length_stack(node *link);
9
10 //create node
11 node *create_node(char* val)
12 {
13     node *p=(node *)malloc(sizeof(node));
14     p->val=val;
15     p->next=NULL;
16     return p;
17 }
18
19 //push 头插法
20 node *push(char* val)
21 {
22     node *p=create_node(val);
23     p->next=head;
24     head=p;
25     return head;
26 }
27
28 //peek
29 char* peek(node *link)

```

```

30 {
31     if(length_stack(link)==0)
32     {
33         printf("此栈为空");
34         return 0;
35     }
36     return head->val;
37 }
38
39 //calculate length_length
40 int length_stack(node *link)
41 {
42     int count=0;
43     while(link)
44     {
45         count++;
46         link=link->next;
47     }
48     return count;
49 }
50
51
52 //pop
53 char* pop(node *link)
54 {
55     node *p;
56     p=head;
57     head=head->next;
58     free(p);
59     return p->val;
60 }
61 //printf_stack
62 void print_stack()
63 {
64     while(head)
65     {
66         printf("%s  ",head->val);
67         head=head->next;
68     }
69 }

```

第六道题

```

1 void transform(double* source, double* dest, size_t n, double (*func)
  (double) ){
2     for( int i=0; i< n; i++ ){
3         dest[i] = func(source[i]);
4     }
5 }

```

第七道题

```

1 typedef int datatype;
2 typedef struct node{
3     datatype id;
4     datatype grade;
5     node* lchild = nullptr;
6     node* rchild = nullptr;
7     node* parent = nullptr;
8     node(datatype id, datatype grade,node* lchild, node* rchild, node*
parent):
9
10    id(id),grade(grade),lchild(lchild),rchild(rchild),parent(parent){}
11
12    node(datatype id, datatype grade): id(id),grade(grade){}
13 }Node;
14
15 void inorder(Node* in){
16     if(in == nullptr)
17         return;
18     inorder(in->lchild);
19     printf("id: %d \t grade: %d \n", in->id, in -> grade);
20     inorder(in->rchild);
21 }

```

第8道题

```

1 class BaseGate{
2 public:
3     BaseGate(){}
4     BaseGate(double length, double width, double
delay):length(length),width(width),delay(delay){}
5 protected:
6     double length;
7     double width;
8     double delay;
9 public:
10     virtual double getArea() =0;
11     virtual double getDelay() = 0;

```

```
12     virtual bool output() = 0;
13
14 };
15
16 class And: public BaseGate{
17 private:
18     bool in1,in2;
19 public:
20     Add(double length, double width, double delay, bool in1, bool in2):
21         BaseGate(length,width,delay), in1(in1), in2(in2){}
22
23     double getArea(){
24         return length * width;
25     }
26
27     double getDelay(){
28         return delay;
29     }
30
31     bool output(){
32         return in1 & in2;
33     }
34
35     /// 实现一个静态重载版本
36     static bool output(bool in1, bool in2){
37         return in1 & in2;
38     }
39 };
40
41 class Or: public BaseGate{
42 private:
43     bool in1,in2;
44 public:
45     Or(double length, double width, double delay, bool in1, bool in2):
46         BaseGate(length,width,delay), in1(in1), in2(in2){}
47
48     double getArea(){
49         return length * width;
50     }
51
52     double getDelay(){
53         return delay;
54     }
55
56     bool output(){
57         return in1 | in2;
58     }
59
60     /// 实现一个静态重载版本
```

```

61     static bool output(bool in1, bool in2){
62         return in1 | in2;
63     }
64 };
65
66
67 class Not: public BaseGate{
68 private:
69     bool in1;
70 public:
71     Not(double length, double width, double delay, bool in1, bool in2):
72         BaseGate(length,width,delay), in1(in1){}
73
74     double getArea(){
75         return length * width;
76     }
77
78     double getDelay(){
79         return delay;
80     }
81
82     bool output(){
83         return ~in1;
84     }
85
86     /// 实现一个静态重载版本
87     static bool output(bool in1){
88         return ~in1;
89     }
90 };
91
92
93 unsigned long simulate(BaseGate *gates[], unsigned int len){
94     unsigned long res = 0;
95     for(int i = 0; i < len ; i ++){
96         gates[i] -> output();
97         res += gates[i] -> getArea();
98     }
99     return res;
100
101 }

```