

第 1 题 (单选)

广域网的英文缩写是 ()。

A WAN

B LAN

C Internet

D WIFI

第 2 题 (单选)

二进制数1011101与下列哪个十六进制数相同 () 。

A (5D)16

B (5B)16

C (D5)16

D (B5)16

第 3 题 (单选)

计算机中最小的信息单位是 () 。

A byte

B KB

C bit

D MB

第 4 题 (单选)

下列哪一个循环会导致死循环 ()。

A for(int k =
0;k<0;k++)

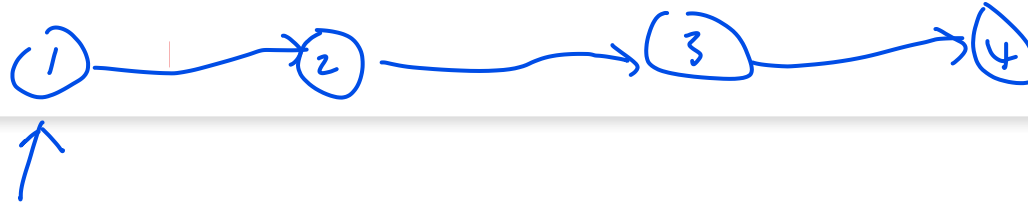
B for(int k =
10;k>10;k--)

 **C** for(int k =
0;k<10;k--)

 **D** for(int k =
0;k>0;k++)

第 5 题 (单选)

对长度位 n 的有序单链表, 若检索每个元素的概率相等, 则顺序检索到表中任一元素的平均检索长度为()。



$$1 + 2 + \dots + n$$
$$\frac{(1+n)n}{2} / n$$

A $n/2$

☒ B $(n+1)/2$

C $(n-1)/2$

D $n/4$

第 6 题 (单选)

把一个复杂的问题分成一个或多个的相同类似的子问题，再把子问题分解成更小的子问题...直到最后的子问题可以简单地直接求解。而原问题的解就是子问题解的并，这种算法思想是（）。

A 动态规划

B 贪心

C 搜索

D 分治

第 7 题 (单选)

以下程序执行完毕后，i和d的值分别为（）。

```
int i, d;  
d = i = 0;  
for(; i <= 7; i+=2)  
    d += i;
```

A 8 10

B 8 12

C 6 10

D 6 12

第 8 题 (单选)

G是一个非连通简单无向图，共有36条边，则该图至少有（）个顶点。

A 10

B 9

C 8

D 7

第 9 题 (单选)

前缀表达式 $-- * 3 5 1 + 1 2$ 的值是 () 。

A 15

B 11

C 3

D 13

第 10 题 （单选）

一棵6节点二叉树的中序遍历为ABDGECF，先序遍历为DBACEGF,后序遍历为（）。

A DGBEFAC

B ABGEFCD

C GBEACFD

D ABCDEFG

第 11 题 (单选)

有正实数构成的数字三角形排列形式如图所示。第一行的数为 $a_{1,1}$ ；第二行的数从左到右依次为 $a_{2,1}, a_{2,2}$ ；... 第 n 行的数为 $a_{n,1}, a_{n,2}, \dots, a_{n,n}$ 。从 $a_{1,1}$ 开始，每一行的数 $a_{i,j}$ 只有两条边可以分别通向下一行的两个数 $a_{i+1,j}$ 和 $a_{i+1,j+1}$ 。用动态规划算法找出一条从 $a_{1,1}$ 向下通到 $a_{n,1}, a_{n,2}, \dots, a_{n,n}$ 中某个数的路径，使得该路径上的数之和达到最大。令 $C_{i,j}$ 是从 $a_{1,1}$ 到 $a_{i,j}$ 的路径上的数的最大和，并且 $C_{i,0} = C_{0,j} = 0$ ，则 $C_{i,j} = ()$ 。

A $\max\{C_{i-1,j-1}, C_{i-1,j}\}$

B $C_{i-1,j-1} + C_{i-1,j}$

C $\max\{C_{i-1,j-1}, C_{i-1,j}\} + 1$

D $\max\{C_{i,j-1}, C_{i-1,j}\}$

第 12 题 (单选)

对于入栈顺序为a, b, c, d, e, f,g的序列, 下列 () 不可能是合法的出栈序列。

A a, b, c, d, e, f,
g

B a, d, c, b, e,
g, f

C a, d, b, c, g, f,
e

D g, f, e, d, c, b,
a

第 13 题 (单选)

下列算法中，（）是稳定的排序算法。

A 快速排序

B 堆排序

C 选择排序

D 插入排序

第 14 题 (单选)

有8本不同的书，其中3本不同的科技书，2本不同的文艺书，3本不同的体育书，将这些书竖排在书架上，则科技书连在一起，文艺书也连在一起的不同排法种数为（）种。

A 720

B 1440

C 2880

D 3600

第 15 题 （单选）

一个抽屉里有20件衬衫，其中4件是蓝的，7件是灰的，9件是红的，则应从中随意取出多少件才能保证有5件是同颜色的？

A 5

B 9

C 12

D 13

第 16 题 (判断)

输入两个正整数a, b。

```
#include <iostream>
#include <algorithm>
using namespace std;
long long myfun(int n, int k) {
    long long now=n, f=n;
    while(k>1) {
        if(k%2==1) {
            k--;
            f*=now;
        }
        if(k) {
            now*=now;
            k/=2;
        }
    }
    return now*f;
}
int main() {
    int a, b;
    cin >> a >> b;
    cout << myfun(a, b);
    return 0;
}
```

当输入2 7时, 会输出128

正确

错误

第 17 题 (判断)

输入两个正整数a, b。

```
#include <iostream>
#include <algorithm>
using namespace std;
long long myfun(int n, int k) {
    long long now=n, f=n;
    while(k>1) {
        if(k%2==1) {
            k--;
            f*=now;
        }
        if(k) {
            now*=now;
            k/=2;
        }
    }
    return now*f;
}
int main() {
    int a, b;
    cin >> a >> b;
    cout << myfun(a, b);
    return 0;
}
```

当输入的a, b值超过某个范围时, 程序会发生运行时错误

正确

错误

第 18 题 (判断)

输入两个正整数a, b。

```
#include <iostream>
#include <algorithm>
using namespace std;
long long myfun(int n, int k) {
    long long now=n, f=n;
    while(k>1) {
        if(k%2==1) {
            k--;
            f*=now;
        }
        if(k) {
            now*=now;
            k/=2;
        }
    }
    return now*f;
}
int main() {
    int a, b;
    cin >> a >> b;
    cout << myfun(a, b);
    return 0;
}
```

程序的时间复杂度是 $O(\log(b))$

正确

错误

第 19 题 (判断)

输入两个正整数a, b。

```
#include <iostream>
#include <algorithm>
using namespace std;
long long myfun(int n, int k) {
    long long now=n, f=n;
    while(k>1) {
        if(k%2==1) {
            k--;
            f*=now;
        }
        if(k) {
            now*=now;
            k/=2;
        }
    }
    return now*f;
}
int main() {
    int a, b;
    cin >> a >> b;
    cout << myfun(a, b);
    return 0;
}
```

当输入10和18时，程序能正常得到一个正整数结果。

正确

错误

第 20 题 (单选)

输入两个正整数a, b。

```
#include <iostream>
#include <algorithm>
using namespace std;
long long myfun(int n, int k) {
    long long now=n, f=n;
    while(k>1) {
        if(k%2==1) {
            k--;
            f*=now;
        }
        if(k) {
            now*=now;
            k/=2;
        }
    }
    return now*f;
}
int main() {
    int a, b;
    cin >> a >> b;
    cout << myfun(a, b);
    return 0;
}
```

第七行if中的条件换成以下哪条语句，程序的功能不变。

A $k|1$

B $!k|1$

C $k\&1$

D $!k\&1$

第 21 题 (判断)

输入n, 随后输入n行指令。

```
#include <iostream>
#include <cstring>
#include <algorithm>
using namespace std;
const int N = 100010;
```

```
    idx ++;
}
void remove(int k) {
    if(ne[k] != -1)
        ne[k] = ne[ne[k]];
}
int main() {
    scanf("%d", &n);
    init();
    while (n --) {
        cin >> op;
        if (op == 'H') {
            scanf("%d", &x);
            add_head(x);
        }
        else if (op == 'D') {
            scanf("%d", &k);
            if(! k) hh = ne[hh];
            else remove(k - 1);
        }
        else if (op == 'I'){
            scanf("%d%d", &k, &x);
```

```
        insert(k - 1, x);  
    }  
}
```

当输入 H x 时，程序会在链表的开头插入 x 元素。

正确

错误

第 22 题 (判断)

输入n，随后输入n行指令。

```
#include <iostream>
#include <cstring>
#include <algorithm>
using namespace std;
const int N = 100010;
... ..
```

```

        idx ++;
    }
    void remove(int k) {
        if(ne[k]!=-1)
            ne[k] = ne[ne[k]];
    }
    int main() {
        scanf("%d", &n);
        init();
        while (n --) {
            cin >> op;
            if (op == 'H') {
                scanf("%d", &x);
                add_head(x);
            }
            else if (op == 'D') {
                scanf("%d", &k);
                if(! k) hh = ne[hh];
                else remove(k - 1);
            }
            else if (op == 'I'){
                scanf("%d%d", &k, &x);
                insert(k - 1, x);
            }
        }
    }
}

```

当输入 D 0 时，程序会删除当前头节点。

正确

错误

第 23 题 (判断)

输入n，随后输入n行指令。

```
#include <iostream>
#include <cstring>
#include <algorithm>
using namespace std;
const int N = 100010;
... ..
```

```

        idx ++;
    }
    void remove(int k) {
        if(ne[k]!=-1)
            ne[k] = ne[ne[k]];
    }
    int main() {
        scanf("%d", &n);
        init();
        while (n --) {
            cin >> op;
            if (op == 'H') {
                scanf("%d", &x);
                add_head(x);
            }
            else if (op == 'D') {
                scanf("%d", &k);
                if(! k) hh = ne[hh];
                else remove(k - 1);
            }
            else if (op == 'I'){
                scanf("%d%d", &k, &x);
                insert(k - 1, x);
            }
        }
        for (int i = hh; i != -1; i = ne[i]) {

```

当输入执行insert或add_head操作N次后，再执行1次remove操作，即可腾出空间继续插入元素。

正确

错误

第 24 题 (单选)

输入n，随后输入n行指令。

```
#include <iostream>
#include <cstring>
#include <algorithm>
using namespace std;
const int N = 100010;
```

```

        idx ++;
    }
    void remove(int k) {
        if(ne[k]!=-1)
            ne[k] = ne[ne[k]];
    }
    int main() {
        scanf("%d", &n);
        init();
        while (n --) {
            cin >> op;
            if (op == 'H') {
                scanf("%d", &x);
                add_head(x);
            }
            else if (op == 'D') {
                scanf("%d", &k);
                if(! k) hh = ne[hh];
                else remove(k - 1);
            }
            else if (op == 'I'){
                scanf("%d%d", &k, &x);
                insert(k - 1, x);
            }
        }
    }
}

```

当输入为:

5

H 1

H 2

H 3

D 0

D 1

输出为:

A 2 1

B 3 2

C 1

D 2

第 25 题 (单选)

输入n，随后输入n行指令。

```
#include <iostream>
#include <cstring>
#include <algorithm>
using namespace std;
const int N = 100010;
```

```

        idx ++;
    }
    void remove(int k) {
        if(ne[k]!=-1)
            ne[k] = ne[ne[k]];
    }
    int main() {
        scanf("%d", &n);
        init();
        while (n --) {
            cin >> op;
            if (op == 'H') {
                scanf("%d", &x);
                add_head(x);
            }
            else if (op == 'D') {
                scanf("%d", &k);
                if(! k) hh = ne[hh];
                else remove(k - 1);
            }
            else if (op == 'I'){
                scanf("%d%d", &k, &x);
                insert(k - 1, x);
            }
        }
    }
}

```

当输入为:

5

H 1

I 1 3

I 2 2

D 2

D 1

输出为:

A 3

B 1 2

C 1

D 1 3

第 26 题 (单选)

输入n, 随后输入n行指令。

```
#include <iostream>
#include <cstring>
#include <algorithm>
using namespace std;
const int N = 100010;
```

```
    idx ++;
}
void remove(int k) {
    if(ne[k] != -1)
        ne[k] = ne[ne[k]];
}
int main() {
    scanf("%d", &n);
    init();
    while (n --) {
        cin >> op;
        if (op == 'H') {
            scanf("%d", &x);
            add_head(x);
        }
        else if (op == 'D') {
            scanf("%d", &k);
            if(! k) hh = ne[hh];
            else remove(k - 1);
        }
        else if (op == 'I'){
            scanf("%d%d", &k, &x);
```

```
    insert(k - 1, x);  
  }  
}
```

在这个简单的链表中，插入与查询的复杂度分别为（假设链表元素有n个）：

A $O(n)$ $O(1)$

B $O(n)$ $O(n)$

C $O(1)$ $O(n)$

D $O(1)$ $O(1)$

第 27 题 (判断)

```
#include <iostream>
#include <string>
using namespace std;
bool check(string s,int start,int end){
    while(start < end){
        if(s[start]!=s[end]){
            return false;
        }
        start++;
        end--;
    }
    return true;
}
int f[100010];
int minCut (string s) {
    int m = s.length();
    if(m == 0){
        return 0;
    }
    for(int i = 0;i <= m;i++){
        f[i] = i-1;
    }
    for (int i = 1;i <= m;i++){
        for (int j = 0;j < i;j++){
            if (check(s,j,i - 1)){
                f[i] = min(f[i],f[j] + 1);
            }
        }
    }
    return f[m];
}
int main() {
    string s;
    cin >> s;
    cout << minCut(s);
}
```

check函数在判断整个字符串s是否为一个回文串。

正确

错误

第 28 题 (判断)

```
#include <iostream>
#include <string>
using namespace std;
bool check(string s,int start,int end){
    while(start < end){
        if(s[start]!=s[end]){
            return false;
        }
        start++;
        end--;
    }
    return true;
}
int f[100010];
int minCut (string s) {
    int m = s.length();
    if(m == 0){
        return 0;
    }
    for(int i = 0;i <= m;i++){
        f[i] = i-1;
    }
    for (int i = 1;i <= m;i++){
        for (int j = 0;j < i;j++){
            if (check(s,j,i - 1)){
                f[i] = min(f[i],f[j] + 1);
            }
        }
    }
    return f[m];
}
int main() {
    string s;
    cin >> s;
    cout << minCut(s);
}
```

输入 aaabaaa 程序会输出0。

正确

错误

第 29 题 (判断)

```
#include <iostream>
#include <string>
using namespace std;
bool check(string s,int start,int end){
    while(start < end){
        if(s[start]!=s[end]){
            return false;
        }
        start++;
        end--;
    }
    return true;
}
int f[100010];
int minCut (string s) {
    int m = s.length();
    if(m == 0){
        return 0;
    }
    for(int i = 0;i <= m;i++){
        f[i] = i-1;
    }
    for (int i = 1;i <= m;i++){
        for (int j = 0;j < i;j++){
            if (check(s,j,i - 1)){
                f[i] = min(f[i],f[j] + 1);
            }
        }
    }
    return f[m];
}
int main() {
    string s;
    cin >> s;
    cout << minCut(s);
}
```

把minCut函数中的

```
for(int i = 0;i <= m;i++){
    f[i] = i-1;
}
```

修改为

```
for (int i = 1; i <= m; i++)
```

修改前后输入aaa，程序得到的结果不会有变化。

正确

错误

第 30 题 (单选)

```
#include <iostream>
#include <string>
using namespace std;
bool check(string s,int start,int end){
    while(start < end){
        if(s[start]!=s[end]){
            return false;
        }
        start++;
        end--;
    }
    return true;
}
int f[100010];
int minCut (string s) {
    int m = s.length();
    if(m == 0){
        return 0;
    }
    for(int i = 0;i <= m;i++){
        f[i] = i-1;
    }
    for (int i = 1;i <= m;i++){
        for (int j = 0;j < i;j++){
            if (check(s,j,i - 1)){
                f[i] = min(f[i],f[j] + 1);
            }
        }
    }
    return f[m];
}
int main() {
    string s;
    cin >> s;
    cout << minCut(s);
}
```

输入 aababccba , 输出为:

A 2

B 3

C 4

D 5

第 31 题 (单选)

```
#include <iostream>
#include <string>
using namespace std;
bool check(string s,int start,int end){
    while(start < end){
        if(s[start]!=s[end]){
            return false;
        }
        start++;
        end--;
    }
    return true;
}
int f[100010];
int minCut (string s) {
    int m = s.length();
    if(m == 0){
        return 0;
    }
    for(int i = 0;i <= m;i++){
        f[i] = i-1;
    }
    for (int i = 1;i <= m;i++){
        for (int j = 0;j < i;j++){
            if (check(s,j,i - 1)){
                f[i] = min(f[i],f[j] + 1);
            }
        }
    }
    return f[m];
}
int main() {
    string s;
    cin >> s;
    cout << minCut(s);
}
```

假如输入的字符串长度为 n ，程序的复杂度为：

A $O(n^2)$

B $O(n^3)$

C $O(n^2 * \log(n))$

D $O(n)$

第 32 题 (单选)

全排列问题，输出 $n(n < 10)$ ，输出 n 的全排列。

如输入：

3

输出：

1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1

使用深度优先搜索解决问题。

```
#include <iostream>  
using namespace std;
```

①处应该填入：

A if(vis[i]) break

B if(vis[i]) continue

C if(!vis[i]) break

D if(!vis[i]) continue

第 33 题 (单选)

全排列问题，输出 $n(n < 10)$ ，输出 n 的全排列。

如输入：

3

输出：

1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1

使用深度优先搜索解决问题。

```
#include <iostream>
using namespace std;
```

②处应该填入：

A $\text{vis}[i] = 0$

B $\text{a}[i] = \text{step}$

C $\text{vis}[i] = 1$

D $\text{a}[i] = \text{step} + 1$

第 34 题 (单选)

全排列问题，输出 $n(n < 10)$ ，输出 n 的全排列。

如输入：

3

输出：

1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1

使用深度优先搜索解决问题。

```
#include <iostream>  
using namespace std;
```

③处应该填入：

A step+1

B step

C i+1

D i

第 35 题 (单选)

全排列问题，输出 $n(n < 10)$ ，输出 n 的全排列。

如输入：

3

输出：

1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1

使用深度优先搜索解决问题。

```
#include <iostream>
```

```
using namespace std;
```

④处应该填入：

A $\text{vis}[i] = 0$

B $\text{a}[i] = \text{step}$

C $\text{vis}[i] = 1$

D $\text{a}[i] = \text{step} + 1$

第 36 题 (单选)

全排列问题，输出 $n(n < 10)$ ，输出 n 的全排列。

如输入：

3

输出：

1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1

使用深度优先搜索解决问题。

```
#include <iostream>
using namespace std;
```

⑤处应该填入：

A ₀

B ₁

C _n

D _{n-1}

第 37 题 (单选)

使用SPFA解决单源最短路问题。

输入一张有向图， n 个点， m 条边，每条边输入 x,y,w ，表示 x 到 y 有一条权值为 w 的边。

①处应该填入：

A $\text{dis}[s] = 0, \text{vis}[s] = 0$

C $\text{dis}[s] = 0, \text{vis}[s] = 1$

B $\text{dis}[s] = -1, \text{vis}[s] = 1$

D $\text{dis}[s] = -1, \text{vis}[s] = 0$

第 38 题 (单选)

使用SPFA解决单源最短路问题。

输入一张有向图， n 个点， m 条边，每条边输入 x,y,w ，表示 x 到 y 有一条权值为 w 的边。

②处应该填入：

A $\text{dis}[v] < \text{dis}[u] + w$

B $\text{dis}[v] > \text{dis}[u] + w$

C $\text{dis}[u] < \text{dis}[v] + w$

D $\text{dis}[u] > \text{dis}[v] + w$

第 39 题 (单选)

使用SPFA解决单源最短路问题。

输入一张有向图， n 个点， m 条边，每条边输入 x,y,w ，表示 x 到 y 有一条权值为 w 的边。

③处应该填入：

A $\text{dis}[u] = \text{dis}[v] - w$

C $\text{dis}[u] = \text{dis}[v] + w$

B $\text{dis}[v] = \text{dis}[u] - w$

D $\text{dis}[v] = \text{dis}[u] + w$

第 40 题 (单选)

使用SPFA解决单源最短路问题。

输入一张有向图， n 个点， m 条边，每条边输入 x,y,w ，表示 x 到 y 有一条权值为 w 的边。

④处应该填入：

A q.push(v), vis[v] = 1

B q.push(v)

C q.push(u), vis[v] = 1

D q.push(u)

第 41 题 (单选)

使用SPFA解决单源最短路问题。

输入一张有向图， n 个点， m 条边，每条边输入 x,y,w ，表示 x 到 y 有一条权值为 w 的边。

⑤处应该填入：

A `e[x].push_back((edge
{y, w}),
e[y].push_back((edge
{x, w}))`



C `e[x].push_back((edge
{w, x}))`



B `e[x].push_back((edge
{y, w}))`



D `e[y].push_back((edge
{x, w}))`



第 42 题 (单选)

使用SPFA解决单源最短路问题。

输入一张有向图， n 个点， m 条边，每条边输入 x,y,w ，表示 x 到 y 有一条权值为 w 的边。

⑥处应该填入：

A 1

B 1e9

C 0

D -1