

一、单项选择题（每题只有一个正确选项，每题 2 分，共 30 分）

1. 在 C++ 中，下列变量名是合法的是（ ）

A. 3f B. 7_8z C. _int D. train.cpp

2. 八进制数 20240512_8 和 21504702_8 的和为（ ）

A. 41745414_8 B. 41745514_8 C. 41745554_8 D. 41744414_8

3. 阅读下述代码，请问将 stu2 的 score 修改为 97 的正确方式是（ ）

```
struct node{  
    string name;  
    char clas;  
    int score;  
}stu1, stu2;
```

A. `stu1->score = 97;` B. `stu2->score = 97;`

C. `stu1.score = 97;` D. `stu2.score = 97;`

4. 对于一个线性表，既要求能够进行较快速地插入和删除，又要求存储结构能反映数据之间的逻辑关系，则应该用（ ）

A. 顺序存储方式 B. 链式存储方式 C. 散列存储方式 D. 以上均可以

5. 在一个单链表中，已知 q 所指结点是 p 所指结点的前驱结点，若在 q 和 p 之间插入结点 s，则执行（ ）

A. `s->next = p->next; p->next = s;` B. `p->next = s->next; s->next = p;`

C. `q->next = s; s->next = p;` D. `p->next = s; s->next = q;`

6. 对于有 n 个顶点、m 条边的无向连通图 ($m > n$)，需要删掉（ ）条边才能使其成为一棵树。

A. $n-1$ B. $m-n$ C. $m-n-1$ D. $m-n+1$

7. 对于一棵具有 n 个结点、度为 4 的树来说，（ ）

- A. 树的高度至多是 $n-3$
- B. 树的高度至多是 $n-4$
- C. 在 i 层上至多有 $4(i-1)$ 个结点
- D. 至少在某一层上正好有 4 个结点

8. 后缀表达式 $9\ 2\ 3\ +\ -\ 10\ 2\ /\ -$ 的值为 ()

- A. 1 B. 2 C. -1 D. -2

9. 运行以下代码片段的行為是 ()

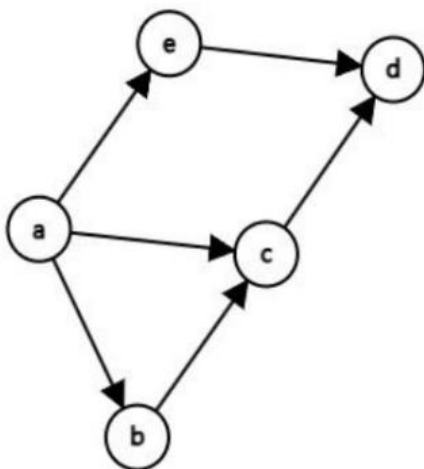
```
int x = 101;
int y = 201;
int *p = &x;
int *q = &y;
p = q;
```

- A. 将 x 的值赋为 201 B. 将 y 的值赋为 101
- C. 将 q 指向 x 的地址 D. 将 p 指向 y 的地址

10. 二进制数 101.11 对应的十进制数是 ()

- A. 6.5 B. 5.5 C. 5.75 D. 5.25

11. 对下图进行拓扑排序，可得不同拓扑序列的个数是 ()



- A. 4 B. 3 C. 2 D. 1

12. 已知一棵二叉树的后序序列为 DABEC，中序序列为 DEBAC，则先序序列为 ()

A. ACBED B. DECAB C. DEABC D. CEDBA

13. 阅读下列代码片段，其中 `sizeof(u)` 的值为 ()

```
union U {  
    bool flag1, flag2, flag3, flag4;  
    int a, b, c;  
    double d;  
    enum E {  
        rank1 = 'A', rank2 = 'B', rank3 = 'C'  
    } e;  
} u;
```

A. 28 B. 27 C. 16 D. 8

14. 关于排序算法，以下哪个选项的说法是错误的 ()

A. 冒泡排序算法是稳定的 B. 简单选择排序是稳定的

C. 简单插入排序是稳定的 D. 归并排序算法是稳定的

15. 10 个苹果放到 7 个篮子里，每个篮子至少有一个苹果，一共有 () 种不同的分配方案。

A. 84 B. 72 C. 56 D. 504

二、阅读程序（程序输入不超过数组或字符串定义的范围；判断题正确填√，错误填×；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

1.

```
01 #include <iostream>  
02 #include <vector>  
03 using namespace std;  
04 int n,k;  
05 vector<vector<int>>res;  
06 vector<int>tmp;  
07 bool vis[20];  
08 void dfs(int layer)  
09 {  
10     if(layer == n+1)  
11     {  
12         res.push_back(tmp);
```

```

13         return;
14     }
15     for(int i=1; i<=n; ++i)
16     {
17         if(vis[i] == 0)
18         {
19             tmp.push_back(i);
20             vis[i] = 1;
21             dfs(layer+1);
22             vis[i] = 0;
23             tmp.pop_back();
24         }
25     }
26 }
27
28 int main()
29 {
30     cin >> n;
31     dfs(1);
32     for(int i=0;i<res.size();++i)
33     {
34         for(int j=0;j<res[i].size();++j)
35         {
36             cout << res[i][j] << " ";
37         }
38         cout << endl;
39     }
40     return 0;
41 }

```

假设输入的 n 是一个整数，且 $1 \leq n \leq 8$ ，完成下面的判断题和单选题：

● 判断题

1. 如果输入 $n = 5$ ，那么程序输出的是自然数 1~5 组成的所有不重复的数字序列（5 的全排列）。（ ）
2. 如果输入 $n = 5$ ，程序第 10 行改写为 $if(layer > n)$ ，程序运行结果不变。（ ）
3. 程序的时间复杂度为 $O(n^2)$ 。（ ）
4. 如果输入 $n = 3$ ，那么就可以将第 7、20、22 行一起删去，程序运行结果不变。（ ）

● 选择题

1. 输入 $n=3$ ，那么程序的输出有（ ）行。

A. 5 B. 6 C. 7 D. 8

2. 输入 $n=4$ ，那么以下序列先被输出的是（ ）。

A. 3 2 4 1 B. 2 4 3 1 C. 3 1 4 2 D. 2 4 1 3

2.

```
#include <iostream>
using namespace std;
const int N = 10000;
int n;
int a[N];
int b[N];
int f(int l, int r, int depth){
    if(l > r) return 0;
    int min = N, mink;
    for(int i = l; i <= r; ++i){
        if(min > a[i]){
            min = a[i];
            mink = i;
        }
    }
    int lres = f(l, mink - 1, depth + 1);
    int rres = f(mink + 1, r, depth + 1);
    return lres + rres + depth * b[mink];
}
int main(){
    cin >> n;
    for(int i = 0; i < n; ++i) cin >> a[i];
    for(int i = 0; i < n; ++i) cin >> b[i];
    cout << f(0, n - 1, 1) << endl;
}
```

● 判断题

1. 如果 a 数组有重复的数字，则程序运行时会发生错误。（ ）

2. 如果 b 数组全为 0，则输出为 0。（ ）

● 选择题

1. 当 $n=100$ 时，最坏情况下，与第 11 行的比较运算执行的次数最接近的是：
（ ）

A. 6 B. 100 C. 5000 D. 600

2. 当 $n = 100$ 时, 最好情况下, 与第 11 行的比较运算执行的次数最接近的是 ()。

A. 5000 B. 6 C. 100 D. 600

3. 当 $n = 10$ 时, 若 b 数组满足, 对任意 $0 \leq i < n$, 都有 $b[i] = i + 1$, 那么输出最大为 ()。

A. 385 B. 383 C. 384 D. 386

4. (4分) 当 $n = 100$ 时, 若 b 数组满足, 对任意 $0 \leq i < n$, 都有 $b[i] = 1$, 那么输出最小为 ()。

A. 582 B. 579 C. 581 D. 580

3.

```
01 #include <iostream>
02 #include <algorithm>
03 using namespace std;
04
05 int n;
06 int a[1010];
07
08 struct node
09 {
10     int h, j, m, w;
11     node(const int _h, const int _j, const int _m, const int _w):
12         h(_h), j(_j), m(_m), w(_w)
13     { }
14
15     node operator+(const node &o) const
16     {
17         return node(
18             max(h, w + o.h),
19             max(max(j, o.j), m + o.h),
20             max(m + o.w, o.m),
21             w + o.w);
22     }
23 };
24
25 node solve1(int h, int m)
26 {
27     if(h > m) return node(-1, -1, -1, -1);
28     if(h == m) return node(max(a[h], 0), max(a[h], 0), max(a[h],
29     0), a[h]);
29     int j = (h + m) >> 1;
```

```

30     return solve1(h, j) + solve1(j + 1, m);
31 }
32
33 int solve2(int h, int m)
34 {
35     if(h > m) return -1;
36     if(h == m) return max(a[h], 0);
37     int j = (h + m) >> 1;
38     int wh = 0, wm = 0;
39     int wht = 0, wmt = 0;
40     for(int i = j; i >= h; i--){
41         wht += a[i];
42         wh = max(wh, wht);
43     }
44     for(int i = j + 1; i <= m; i++){
45         wmt += a[i];
46         wm = max(wm, wmt);
47     }
48     return max(max(solve2(h, j), solve2(j + 1, m)), wh + wm);
49 }
50
51 int main()
52 {
53     cin >> n;
54     for(int i = 1; i <= n; i++) cin >> a[i];
55     cout << solve1(1, n).j << endl;
56     cout << solve2(1,n) << endl;
57     return 0;
58 }

```

● 判断题

1. 程序会输出两个数，且这两个数相等。（ ）
2. 第 27 行有可能执行两次及以上。（ ）
3. 第 35 行有可能执行两次及以上。（ ）
4. 输入“5 -10 11 -9 5 -7”，第一行和第二行都输出“7”。（ ）

● 选择题

1. **solve2(1, n)**的时间复杂度为（ ）。

A. $\theta(\log n)$ B. $\theta(n)$ C. $\theta(n \log n)$ D. $\theta(n^2)$

2. 输入“10 -3 2 10 0 -8 9 -4 -5 9 4”，输出的第二行为（ ）

三、完善程序（单选题，每小题 3 分，共计 30 分）

1.（买材料）工厂负责人需要采购 n 份材料，现在有 m 个供应商，他们供应材料的单价为 p_i ，最多能供应的数量为 a_i ，求工厂负责人买到 n 份材料所需的最小费用。

输入格式：

第一行两个整数 n, m ， n 表示工厂所需材料个数， m 表示供应商个数

接下来 m 行，没行两个整数 p_i, a_i ，表示第 i 个供应商材料的单价，和供应商 i 最多能供应的材料数量

题目保证工厂负责人总能买够所需的材料。

输出格式：

输出工厂负责人买材料所需的最小费用。

输入 输出

100 5 630

5 20

9 40

3 10

8 80

6 30

说明： $0 \leq n, a_i \leq 2 \times 10^6, 0 \leq m \leq 5000, 0 \leq p_i \leq 1000$

试补全程序。

```
#include <iostream>
#include <algorithm>
using namespace std;
const int N = 5050;
struct node
{
    int price,num;
    bool operator <(const node& b)const{
        return ①;
    }
}a[N];
int main()
{
    int n,m;
    cin >> n >> m;
    for(int i = 1; i <= m; ++i)
```



```

cin >> a[i].price >> a[i].num;
if(②)
{
    cout << 0;
    return 0;
}
③;
int ans = 0;
int cnt = 0;
for(int i = 1; i <= m; ++i)
{
    if(④ > n)
    {
        int k = ⑤;
        ans += k*a[i].price;
        break;
    }
    else
    {
        cnt += a[i].num;
        ans += a[i].price*a[i].num;
    }
}
cout << ans;
}

```

34.①处应填 ()

A. price < b.price B. price > b.price C. num < b.num D. num > b.num

35.②处应填 ()

A. n == 0 B. n == 1 C. m == 0 D. m == 1

36.③处应填 ()

A. sort(a+1, a+1+n) B. sort(a, a+n) C. sort(a+1, a+1+m) D. sort(a, a+m)

37.④处应填 ()

A. ans + a[i].price B. cnt + a[i].price C. ans + a[i].num D. cnt + a[i].num

38.⑤处应填 ()

A. m - ans B. n - cnt C. n - ans D. m - cnt

2. (挑数求值) 给定一个长度为 n 的数组，从这个数组中挑选一些值，问取出的数产生的平均数最大能是多少？同样的，中位数最大能是多少？

注意，在你挑选时，需要满足：任意相邻两个数里面最少选一个。

并且，本题中的中位数定义为：如果是奇数，就是从小到大排序后最中间的那个数字；如果是偶数，就是从小到大排序后中间靠左的元素（中左元素、上中位数）。例如：1 2 3 的中位数是 2，1 2 3 4 的中位数是 2。

题目保证：

- $2 \leq N \leq 10^5$
- $1 \leq A_i \leq 10^9$
- 所有的输入均为整数

输入	输出
6	4
2 1 2 1 1 10	2

试补全程序。

```
#include <iostream>
using namespace std;
const int N = 1e5+6;
int n;
int a[N];
double dp[N][2];

bool check1(double x){
    dp[0][0] = 0.0;
    dp[0][1] = 0.0;
    for(int i = 1; i <= n; ++i){
        dp[i][0] = dp[i-1][1];
        dp[i][1] = max(dp[i-1][0], dp[i-1][1]) + (①);
    }
    if(max(dp[n][0], dp[n][1]) >= 0) return true;
    else return false;
}
```

```
bool check2(int x){
    dp[0][0] = 0;
    dp[0][1] = 0;
    for(int i = 1; i <= n; ++i){
        dp[i][0] = dp[i-1][1];
        int num;
        if(②) num = 1;
        else num = -1;
        dp[i][1] = max(dp[i-1][0], dp[i-1][1]) + (③);
    }
}
```

```

    }
    if(max(dp[n][0], dp[n][1]) > 0) return true;
    else return false;
}

int main()
{
    cin >> n;
    for(int i=1; i <= n; ++i) cin >> a[i];
    double l = 0, r = 1000000000.0;
    while(r - l >= 1e-6){
        double mid = (l + r) / 2.0;
        if(check1(mid)) ④;
        else r = mid;
    }
    int le = 0, ri = 1000000000;
    while(le <= ri){
        int mid = le + ri >> 1;
        if(check2(mid)) ⑤;
        else ri = mid - 1;
    }
    cout << r << endl;
    cout << ri;
    return 0;
}

```

39. ①处应填 ()

A. $a[i] - x$ B. $a[i]$ C. $a[i] + x$ D. x

40. ②处应填 ()

A. $a[i] > x$ B. $a[i] \leq x$ C. $a[i] \geq x$ D. $a[i] < x$

41. ③处应填 ()

A. $num + x$ B. $num - x$ C. $num + 1$ D. num

42. ④处应填 ()

A. $l = mid$ B. $l = mid + 1$ C. $l = mid - 1$ D. $l = l + mid / 2.0;$

43. ⑤处应填 ()

A. $le = le + mid / 2$ B. $le = mid$ C. $le = mid - 1$ D. $le = mid + 1$