1.32 位 int 类型的存储范围是（　　）

A. -2147483647 ~ +2147483647

B. -2147483647 ~ +2147483648

C. -2147483648 ~ +2147483647

D. -2147483648 ~ +2147483648

题解：C

int 的范围为"-2^{31}~$2^{31}-1$",算出来的结果是-2147483648~2147483647

2. 计算$(14_8-1010_2)*D_{16}-1101_2$的结果，并选择答案的十进制值：（　　）

A.13

B.14

C.15

D.16

题解：A

原式化成十进制后为(12-10)*13-13，计算后得 13

3. 某公司有 10 名员工，分为 3 个部门：A 部门有 4 名员工，B 部门有 3 名员工、C 部门有 3 名员工。现需要从这 10 名员工中选出 4 名组成一个工作组，且每个部门至少要有 1 人。问有多少种选择方式？（　　）

A.120

B.126

C.132

D.238

题解：B

A 选 2 人，BC 各一人:$C_4^2*C_3^1*C_3^1=54$

B 选 2 人，AC 各一人:$C_3^2*C_4^1*C_3^1=36$

C 选 2 人，AB 各一人:$C_3^2*C_4^1*C_3^1=36$

加一起得 126

4. 以下哪个序列对应数组 0 至 8 的 4 位二进制格雷码（Gray code）？

A.0000，0001，0011，0010，0110，0111，0101，1000

B.0000，0001，0011，0010，0110，0111，0100，0101

C.0000，0001，0011，0010，0100，0101，0111，0110

D.0000，0001，0011，0010，0110，0111，0101，0100

题解：D

根据格雷码定义推算即可

5. 记 1Kb 位 1024 字节（byte），1MB 位 1024KB，那么 1MB 是多少二进制位（bit）？

A.1000000

B.1048576

C.8000000

D.8388608

题解：D

一个字节占用 8 个 bit 位，因此 1MB 一共占用 1024*1024*8=8388608 个 bit 位

6. 以下哪个**不是** C++中的基本数据类型？

A. Int

B. float

C. struct

D. char

题解：C

struct 不是基本数据类型

7. 以下哪个不是 C++中的循环语句？

A. for

B. while

C. do-while

D. repeat-untill

题解：D

Repeat-untill 是 Pascal、lua 等语言中的直到循环语句，C++并不支持

8. 在 C/C++中，(char)('a'+13)与下面的哪一个值相等（　　）

A. 'm'

B. 'n'

C. 'z'

D. '3'

题解:B

'a'的 ASCII 码为 97,'a'+13 为 110，ASCII 为'n'

9. 假设有序表中有 1000 个元素，则用二分法查找元素 x 最多需要比较（　　）次

A.25

B.10

C.7

D.1

题解：B

$2^{10} = 1024 > 1000$故选 B

10. 下面哪一个不是操作系统名字（　　）

A. Notepad

B. Linux

C. Windows

D. macOS

题解：A

其他三个都是操作系统

11. 在无向图中，所有顶点的度数之和等于（　　）

A. 图的边数

B. 图的边数的两倍

C. 图的定点数

D. 图的定点数的两倍

题解：B

一条边贡献两个度，因此所有顶点度数之和等于边数的两倍

12. 已知二叉树的前序遍历为[A,B,D,E,C,F,G],中序遍历为[D,B,E,A,F,C,G],求二叉树的后序遍历的结果是（　　）

A. [D,E,B,F,G,C,A]

B. [D,E,B,F,G,A,C]

C. [D,B,E,F,G,C,A]

D. [D,E,B,F,G,A,C]

题解：A

还原二叉树即可

13. 给定一个空栈，支持入栈和出栈操作。若入栈操作的元素依次是 1 2 3 4 5 6,其中 1 最先入栈，6 最后入栈，下面哪种出栈顺序是不可能的（　　）

A.6 5 4 3 2 1

B.1 6 5 4 3 2

C.2 4 6 5 3 1

D.1 3 5 2 4 6

题解：D

A 的顺序为：1 进 2 进 3 进 4 进 5 进 6 进，6 出 5 出 4 出 3 出 2 出 1 出

B 的顺序为：1 进，1 出 2 进 3 进 4 进 5 进 6 进，6 出 5 出 4 出 2 出 2 出

C 的顺序为：1 进 2 进，2 出，3 进 4 进，4 出，5 进 6 进，6 出 5 出 3 出 1 出

D 无法实现

14. 有 5 个男生和 3 个女生站成一排，规定 3 个女生必须相邻，问有多少种不同的排列方式？

A. 4320 种

B. 5040 种

C. 3600 种

D. 2880 种

题解：A

捆绑法，将三个女生绑在一起方法总数为A_3^3，和所有男生站一起总数为$A_3^3 * A_6^6 = 4320$

15. 编译器的主要作用是什么（　　）？

A. 直接执行源代码

B. 将源代码转换为机器代码

C. 进行代码调试

D. 管理程序运行时的内存

题解：B

(1)

```
1   #include <iostream>
2   using namespace std;
3
4   bool isPrime(int n) {
5       if (n <= 1) {
6           return false;
7       }
8       for (int i = 2; i * i <= n; i++) {
9           if (n % i == 0) {
10              return false;
11          }
12      }
13      return true;
14  }
15
16  int countPrimes(int n) {
17      int count = 0;
18      for (int i = 2; i <= n; i++) {
19          if (isPrime(i)) {
20              count++;
21          }
22      }
23      return count;
24  }
25
26  int sumPrimes(int n) {
27      int sum = 0;
28      for (int i = 2; i <= n; i++) {
29          if (isPrime(i)) {
30              sum += i;
31          }
32      }
33      return sum;
34  }
35
35  int main() {
37  int x;
38  cin >> x;
39  cout << countPrimes(x) << " " << sumPrimes(x) << endl;
40  return 0;
41  }
```

题解：√×√BA
题目中 3 个函数，其中 isPrime(x)为判断质数的函数，countPrimes(x)为判断 1~x 中有多少个质数的函数，sumPrimes(x)为求 1~x 中所有质数的和

判断题：

16.当输入为“10”时，程序的第一个输出为“4”，第二个输出为“17”。（ ）
题解：√
直接计算即可

17.若将 isPrime(i)函数种的条件改为 i<=n/2,输入“20”时，countPrimes(20)的输出将变为“6”（ ）
题解：×
应为 8

18.sumPrimes 函数计算的是从 2 到 n 之间的所有素数之和（ ）
题解：√

单选题

19.当输入为“50”时，sumPrimes(50)的输出为（ ）
A.1060
B.328
C.381
D.275
题解：B
1~50 质数的和为 328

20.如果将 for(int i=2;i*i<=n;i++)改为 for(itn i=2;i<=n;i++),输入“10”时，程序的输出（ ）
A.将不能正确计算 10 以内素数个数及其和
B.仍然输出“4”和“17”
C.输出“3”和 10
D.输出结果不变，但余小宁时间更短
题解：A
加上等号以后所有数都不满足条件

（2）

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

int compute(vector<int> &cost) {
    int n = cost.size();
    vector<int> dp(n + 1, 0);
    dp[1] = cost[0];
    for (int i = 2; i <= n; i++) {
        dp[i] = min(dp[i - 1], dp[i - 2]) + cost[i - 1];
    }
    return min(dp[n], dp[n - 1]);
}

int main() {
    int n;
    cin >> n;
    vector<int> cost(n);
    for (int i = 0; i < n; i++) {
        cin >> cost[i];
    }
    cout << compute(cost) << endl;
    return 0;
}
```

题解：√××ABA

判断题：

21.当输入的 cost 数组为{10，15，20}时，程序的输出为 15（　　）
题解：√
直接根据代码计算即可

22.如果将 dp[i-1]改为 dp[i-3]，程序可能会产生编译错误（　　）
题解：×
不会产生编译错误，因为数组下标为负数，会产生运行错误

23.（2 分）程序总是输出 cost 数组种的最小的元素（　　）
题解：×

·单选题

24．当输入的 cost 数组为｛1,100,1,1,1,100,1,1,100,1｝时，程序的输出为（）。

A."6"

B."7"

C."8"

D."9"

题解 A

直接根据代码计算即可

25．（4 分）如果输入的 cost 数组为｛10,15,30,5,5,10,20｝，程序的输出为()

A."25"

B."30"

C."35"

D."40"

题解：B

直接根据代码计算即可

26.若将代码中的 min(dp[i-1],dp[i-2])+cost[i-1]修改为 dp[i-1]+cost[i-2]，输入 cost 数组为{5,10,15}时，程序的输出为（）

A."10"

B."15"

C."20"

D."25"

题解：A

直接根据代码计算即可

（3）

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

int customFunction(int a, int b) {
        if (b == 0) {
        return a;
        }
        return a + customFunction(a , b - 1);
}

int main() {
int x, y;
cin >> x >> y;
int result = customFunction(x, y);
cout << pow(result, 2) << endl;
return 0;
}
```

题解：× √ √BCD
customFunction 函数为计算(b+1)个 a 相加的结果
最后输出是 customFuntion 的平方

判断题：
27. 当输入为“2 3”时，customFunction（2,3）的返回值为“64”。（ ）
题解：×
这个函数的返回值应该为 8

28. 当 b 为负数时，customFunction（a，b）会陷入无限递归。（ ）
题解：√
B 为负数递归无终止条件

29. 当 b 的值越大，程序的运行时间越长。（ ）
题解：√
b 越大，递归次数越多

单选题

30，当输入为“5 4”时，customFunction（5,4）的返回值为(　　)。

A.5

B.25

C.250

D.625

题解 B

5 个 5 相加等于 25

31．如果输入 x = 3 和 y = 3，则程序的最终输出为（）

A."27”

B."81”

C."144”

D."256

题解：C

customFunction(3,3)的返回值为 12

$12^2 = 144$

32．（4 分）若将 customFunction 函数改为“return a ＋ customFunction（a-1，b-1）；并输入“3 3”，则程序的最终输出为（）。

A.9

B.16

C.25

D.36

题解：D

直接根据代码计算即可

(1) (判断平方数) 问题：给定一个正整数 n，判断这个数 是不是完全平方数，即存在一个正整数 x 使得 x 的平方等于 n

试补全程序

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

bool isSquare(int num){
        int i =      (1)      ;
        int bound =      (2)      ;
        for(;i<=bound;++i){
                if(      (3)      ){
                        return      (4)      ;
                }
        }
        return      (5)      ;
}
int main(){
        int n;
        cin>>n;
        if(isSquare(n)){
                cout<<n<<" is a Square number"<<endl;
        }else{
                cout<<n<<" is not a Square number"<<endl;

        }
```

33．①处应填（）

A. 1　B.　2. C.　3.　D.4

题解：A

34．②处应填（）

A.　(int) floor(sqrt(num)-1)

B.　(int)floor(sqrt(num))

C. floor(sqrt(num/2))-1

D. floor(sqrt(num/2))

题解：B

35．③处应填（）

A. num=2*i

B. num== 2*i

C. num=i*i

D. num==i*i

题解：D

36．④处应填（）

A. num= 2*i

B. num==2*i

C. true

D. false

题解：C

37．⑤处应填（）

A. num= i*i

B. num!=2*I

C. true

D. False

题解：D

2）（汉诺塔问题）给定三根柱子，分别标记为 A、B 和 C。初始状态下，柱子 A 上有若干个圆盘，这些圆盘从上到下按从小到大的顺序排列。任务是将这些圆盘全部移到柱子 c 上，且必须保持原有顺序不变。在移动过程中，需要遵守以不规则：

1．只能从一根柱子的顶部取出圆盘，并将其放入另一根柱子的顶部。

2．每次只能移动一个圆盘

3．小圆盘必须始终在大圆盘之上。

试补全程序

```
#include <bits/stdc++.h>
using namespace std;

void move(char src, char tgt) {
    cout << "从柱子" << src << "挪到柱子上" << tgt << endl;
}
void dfs(int i, char src, char tmp, char tgt) {
    if(i ==    (1)    ) {
        move(    (2)    );
        return;
    }
    dfs(i-1,    (3)    );
    move(src, tgt);
    dfs(    (5)    ,    (4)    );
}

int main() {
    int n;
    cin >> n;
    dfs(n, 'A', 'B', 'C');
}
```

38．①处应填（）

A.0

B.1

C.2

D. 3

题解：B

39．②处应填（）

A. src,tmp

B. src,tgt

C. tmp,tgt

D.tgt,tmp

题解：B

40．③处应填（）

A. src,tmp,tgt

B. src, tgt, tmp

C. tgt, tmp, src

D. tgt, src, tmp

题解：B

41．④处应填（）

A. src, tmp, tgt

B. tmp,src, tgt

C. src, tgt,tmp

D. tgt,src,tmp

题解：B

42．⑤处应填（）

A. 0

B. 1

C. i-1

D. i

题解：C