Started on	Thursday, 21 September 2023, 5:06 PM
State	Finished
Completed on	Thursday, 21 September 2023, 5:21 PM
Time taken	15 mins
Grade	**6.00** out of 10.00 (**60**%)

Question **1**

Incorrect

Mark 0.00 out of 1.00

What does the following function print for n = 25?

```
void test(int n) {
    if (n == 0)
        return;
    printf("%d", n%2);
    test(n/2);
}
```

- a. 11111
- b. 11001 ✖
- c. 00000
- d. 10011

The correct answer is:
10011

Question **2**

Correct

Mark 1.00 out of 1.00

Match the following with respect to the following program segment:

int arr[3][3] = {{2,4,6}, {9,1,10}, {16, 64, 5}};

*(arr[0]) | *(arr[2])	18 ✔
arr[0][1] ^ arr[0][2]	2 ✔
**arr < *(*arr+2)	1 ✔
**arr	2 ✔
*(arr[1] + 1) | arr[1][2]	11 ✔

The correct answer is:
*(arr[0]) | *(arr[2]) → 18,

arr[0][1] ^ arr[0][2] → 2,

**arr < *(*arr+2) → 1,

**arr → 2,

*(arr[1] + 1) | arr[1][2] → 11

Question **3**

Correct

Mark 1.00 out of 1.00

```
int main() {
   char *p = "Programming";
   printf("%c", *& * & * p);
   return 0;
}
```

- (●) a. P ✔
- ( ) b. Runtime Error
- ( ) c. Programming
- ( ) d. Garbage value

The correct answer is:
P

Question **4**

Correct

Mark 1.00 out of 1.00

What will be the output produced by the following C code:

```
int main() {
  int arr[5][5];
  printf("%d", ((arr == *arr) && (*arr == arr[0]) ));
  return 0;
}
```

- a. 1 ✔
- b. 0
- c. -1
- d. 2

The correct answer is:
1

Question **5**

Correct

Mark 1.00 out of 1.00

Consider the following recursive function fun(x, y). What is the value of fun(4, 3)?

```
int fun(int x, int y) {
  if (x == 0)
  return y;
  return fun(x - 1, x + y);
}
```

- a. 13 ✔
- b. 12
- c. 10
- d. 9

The correct answer is:
13

Question **6**

Correct

Mark 1.00 out of 1.00

Which of the following is the correct way for declaring a float pointer?

- a. float ptr
- b. float *ptr ✔
- c. None of the above
- d. *float ptr

The correct answer is:
float *ptr

Question **7**

Incorrect

Mark 0.00 out of 1.00

What is the output of the following program?

```
#include <stdio.h>
void func(int *a, int *b)
{
   a = b;
   *a = 2;
}
int i = 0, j = 1;
int main()
{
   func(&i, &j);
   printf("%d %d", i, j);
   return 0;
}
```

- a. 2 1
- b. 0 2
- c. 0 1
- d. 2 2 ✖

The correct answer is:
0 2

Question **8**

Incorrect

Mark 0.00 out of 1.00

What is the problem with following code?

```
#include<stdio.h>
int main() {
   int *ptr = (int *)malloc(sizeof(int));
   ptr = NULL;
   free(ptr);
}
```

- a. Compiler Error: free can't be applied on NULL pointer
- b. Dangling Pointer ✖
- c. Memory Leak

The correct answer is:
Memory Leak

Question **9**

Correct

Mark 1.00 out of 1.00

What is the return type of the function with prototype: "int func(char x, float v, double t);"

- a. float
- b. char
- c. int ✔
- d. double

The correct answer is:
int

Question **10**

Incorrect

Mark 0.00 out of 1.00

What happens when one assigns a value to an element of array whose subscript exceeds the size of the array?

- a. Nothing, it is done all the time ✖
- b. The element is set to zero
- c. Other data may be overwritten
- d. Compiler error

The correct answer is:
Other data may be overwritten

◄ Midsem-Theory-Section-A

Jump to...

Endsem Theory ►

Voting

View as PDF

Problem Description

The Elections for the new academic year are in full swing!

There are N students in the college and two candidates for the current elections (where N is guaranteed to be an odd number). The students cast their votes in the form of $0s$ or $1s$, where the i^{th} vote (1≤i≤N $1 \leq i \leq N$) being 1 represents a vote to candidate A and it being 0 represents a vote for candidate B.

A candidate wins the elections if they have the strict majority. That is, if the number of votes cast for them is strictly greater than the number of votes cast for the other candidate.

Output `Yes` if candidate A wins the elections and `No` otherwise.

Input Format

The first line of input contains N, the number of students. Then, N line follows.

The i^{th} line contains a single integer $V_i = \{0, 1\}$ where 1 represents a vote for candidate A and 0 represents a vote for candidate B.

Input constraints

- $1 \leq N \leq 99$
- N is an odd number
- $V_i = \{0, 1\}$

Output Format

Output `Yes` if candidate A wins the elections and `No` otherwise.

Sample Input 1

```
3
1
0
1
```

Copy

Sample Output 1

Submit solution

My submissions
All submissions
Best submissions

Points: 100 (partial)
Time limit: 1.0s
Memory limit: 256M

Allowed languages
C

Explanation

There are 2 votes for candidate A and 1 vote for candidate B. Therefore, candidate A has the majority and wins the elections.

Sample Input 2

```
5
0
0
1
0
1
```

Copy

Sample Output 2

```
No
```

Copy

Explanation

There are 2 votes for candidate A and 3 votes for candidate B. Therefore, candidate B has the majority and wins the elections.

Clarifications

Request clarification

No clarifications have been made at this time.

Jump back

View as PDF

Problem Description

There are N trees in the forest, numbered Tree 1, Tree 2,..., Tree N.

Tree 1, is a special tree which is home to the entire forest. Each tree i ($2 \leq i \leq N$), has a parent tree T_i from which it grew. It is guaranteed that $T_i < i$.

You are currently on tree N and want to reach tree 1. However, from a given tree, you can only jump to its parent tree (unless you are already on tree 1, in which case you have already reached the destination).

For example, consider the input $[1124]$ which means,
$T_2 = 1, T_3 = 1, T_4 = 2, T_5 = 4$ (note that tree 1 has no parent tree). Starting from tree 5, the only option is to jump to tree 4, its parent. Similarly, from tree 4, you jump to tree 2. Finally, from tree 2 you jump to tree 1, taking a total of 3 jumps to go from tree 5 to tree 1.

Find the number of jumps that you need to make in order to reach tree 1 starting from tree N. It can be shown that you can reach it in a finite number of jumps.

Input Format

The first line of input contains a single integer N, the number of trees.

The following line contains $N-1$ space separated integers, $T_2, T_3, \ldots, T_N$ where T_i denotes the parent of the i^{th} tree.

Input constraints

- $2 \leq N \leq 50$
- $1 \leq T_i < i$ ($2 \leq i \leq N$)

Output Format

Output a single integer denoting the number of jumps required to go from tree N to tree 1.

Sample Input 1

```
3
1 2
```

Copy

Submit solution

My submissions
All submissions
Best submissions

Points: 100 (partial)
Time limit: 1.0s
Memory limit: 256M

Allowed languages
C

```
2
```

Copy

Sample Input 2

```
5
1 1 2 4
```

Copy

Sample Output 2

```
3
```

Copy

Clarifications

Request clarification

No clarifications have been made at this time.

Max AND

View as PDF

Problem Description

In this problem, we use the symbol [^] to denote the **bitwise AND** operation, and the symbol [v] to denote the **bitwise OR** operation.

Given an array of integers A of length N, $A_1, A_2, \ldots, A_N$. Define a "set operation" as the following:

- Select any index i where $1 \leq i \leq N$ and set the j^{th} bit of A_i where j is any integer between 0 and 30 inclusive. In other words, replace A_i with A_i [v] 2^j.

You are also given a non-negative integer K. You can perform **at most** K operations on the given array. After doing so, output the maximum possible value of A_1 [^] A_2 [^] $\ldots$ [^] A_N.

Input Format

The first line of input contains two space separated integers N and K, denoting the length of the array and the maximum number of operations that can be performed respectively.

The second line contains N space separated integers $A_1, A_2, \ldots, A_N$ that denote the initial array.

Input constraints

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq K \leq 10^9$
- $0 \leq A_i < 2^{31}$

Output Format

Output the maximum possible value of the bitwise AND of the entire array after performing at most K operations.

Sample Input 1

```
3 2
2 1 1
```

Copy

Submit solution

My submissions
All submissions
Best submissions

Points: 100 (partial)
Time limit: 3.0s
Memory limit: 256M

Allowed languages
C

```
2
```
Copy

Explanation

We set the $j = 1$ bit (that is, the bit corresponding to 2^1) for A_2 and A_3. This requires $2 \leq K$ operations. After doing so, the array now looks like $[2, 3, 3]$, giving us a bitwise AND value of 2 which can be shown is the maximum possible value for the given input.

Sample Input 2

```
7 0
4 6 6 28 6 6 12
```
Copy

Sample Output 2

```
4
```
Copy

Explanation

Here, $K = 0$. So, we cannot perform any operations. Hence, the bitwise AND of the initial array, 4 is our final answer.

Clarifications

Request clarification

No clarifications have been made at this time.