



iKnight Coders

Participation: 3 Member team

Eligibility: All students with a **valid identity card** of their respective educational institutes and also their Aranya registration cards are eligible to participate.

Medium: C/C++ (to be compiled using GCC) & Java

Prelims

Date: 16th September 2008

Time: 5:30 – 6:30 PM

Results for the prelim round will be declared by 7:30 – 8:00 PM

After the dinner break, the finals of the intra college event start at 9:30 PM. The finalists should be present in the labs latest by 9:15 PM after which the new entrants will be rejected.

Prizes:

1. The prelims will select the **30 finalist** teams.
2. The team that stands first in the prelims will be given a **prize worth Rs 750**.

Prelims would consist of Multiple Choice questions/Short Programs to be written in C/C++. The questions will be based on C/C++/Aptitude/Computer General Knowledge. It will be time based.

Sample Questions:

1. What will be the output of the following program:

```
int main()
{
    cout<<"Hello";
    return 0;
}
```

- a. Hello
- b. hello
- c. Hillo
- d. Other: _____



The answer is: a

2. Which integer precedes 6?
- a. 2
 - b. 6
 - c. 5
 - d. Other: _____

The answer is: c

3. Who is the father of Computers?
- a. Osama Bin Laden
 - b. Bill Gates
 - c. Charles Babbage
 - d. Other: _____

The answer is: c

4. Write a program to print the first 10 natural numbers separated by a space.

The answer is:

```
#include<iostream.h>
void main()
{
    //Acceptable to Turbo C++ compiler
    for (int i=1;i<=10;i++)
    {
        cout<<i<<" ";
    }
}
```

Note: The allowed compilers include Borland C++, GNU GCC/G++, Turbo C++ for the prelims.

- These questions will be on paper and in a restricted time.
- The decision of the judges will be final.
- No cutting, erasing or overwriting of the answers will be allowed.



Mains

Date: 16th September 2008

Time: 9:30 – 2:30 PM

Prizes:

1. The top ten teams in the finals will be given wild card entries in the event Knight Coders.
2. First **Prize worth Rs. 3000.**
3. Second **Prize worth Rs. 2000.**
4. **A cash prize to the fastest coder with the first correct solution of Rs 250.**

The mains will consist of questions in the international format of programming questions and will be evaluated only by the PC² Software provided.

1. **Scoring:** The algorithm used Rank and “Score” (Penalty Points) is the one used in the ACM ICPC World Finals, which is as follows:
 - a. Teams are ranked according to the number of problems solved; a team solving more problems is always ranked higher than a team solving fewer problems.
 - b. Within a group of teams solving the same number of problems, teams are ranked by increasing “Penalty Points” (that is, the team with the lowest number of Penalty Points is ranked highest within the group). Teams only accrue Penalty Points for problems which the team has **solved**; unsolved problems do not affect the scoring in any way.
 - c. Teams accrue Penalty Points for solved problems in two ways:(**Penalty Points will come into picture only in the case of a tie breaker**)
 - i. One point for each minute elapsed from the start of the contest until the problem was solved (the time of SUBMISSION is counted as the “time solved”; it does not matter how long it took the Judges to judge it).
 - ii. A specific number of penalty points for each INCORRECT submission submitted to the Judges *prior to a correct solution* for the problem (runs submitted after a correct solution is not counted in the scoring). The number of penalty points for each incorrect submission prior to solving a problem is 20.
 - d. If two or more teams have the same number of solved problems and exactly the same number of Penalty Points, ties are broken in favor of the team with the earliest time of the last correct submission (that being the time when the team “finished” the contest).
2. Only the GNU GCC compiler is allowed, e.g. **Dev C++** which can be downloaded at: <http://www.bloodshed.net/dev/devcpp.html>. Note that it does not support void main(you have to use int main), clrscr(), strev() of turbo C++ compiler. Other language allowed is **Java(JDK 1.5)**.
 - a. The exe file should be exactly same in all cases.
 - b. The spellings as well as the input and output format should match 100% otherwise the software will declare an error.
3. Note:



- a. You should submit only the source code. The explanation (if reqd.) should be present as the comment.
 - b. The team no. and the team name along with the member usernames, roll nos., email addresses and preferably a contact no. (For quick references) should be present in the comment section on the top.
4. No indiscipline will be tolerated.
 5. Any team found copying, cheating or exchanging notes will be disqualified.
 6. The decision of the organizers will be final.

Sample Question:

Q1 According to Darwinism, there is competition among all living beings and the fittest of the set survives. He says that all living species fight for the available resources and only a few members outlive the others and that is why maybe the dinosaurs got extinct. Consider a case just opposite.

Life has just begun on earth. Only one pair lived - Adam and Eve. They give rise to two pairs, each of which produces another two pairs. There is no fight and there are limitless resources. People die only of age. Everyone is able to see their grandchildren but none sees their great grandchildren. Only three generations live at a time.

The first input line will be the number of test cases followed by an input for each test case. Each case will consist of the generation that is just born i.e. n and you have to find out the number of pairs that inhabit the earth.

Sample Input:

2
3
5

Sample Output

7
28

Constraints

$3 < n < 14$



```
// Sample Solution
#include<math.h>
#include<iostream>
using namespace std; //The iostream headers have to be used this way in Dev C++
int main() //Not void
{
    int noOfCases;           //The stuff outside the for loop is normally
    cin>>noOfCases;         //the same in all questions
    for(int i=0;i<noOfCases; i++)
    {
        int generation;
        cin>>generation;
        int thisgen,prevgen,oldergen,totalpairs;
        thisgen = pow(2,generation-1);
        prevgen = pow(2,generation-2);
        oldergen = pow(2,generation-3);
        totalpairs = thisgen + prevgen + oldergen;
        cout<<totalpairs;
        cout<<endl;         //End of the test case
    }
    return 0;
}
```

