```
25 5 3
1 2 6 10
You find yourself in a cave... with diamonds! Shame you don't have an iron pickaxe.
2 1 3 7
You look around and realize you've found a stronghold.
3 2 4 8
Wow! A room full of cute pigs! *Oink Oink*
4 3 5 9
An armor stand in corner is decked out in diamond armor.
5 4 6 10
You've found a rare mooshroom... get some soup!
6 1 5 7
You enter a room full of abandoned chests and furnaces.
7 2 6 8
You see a bunch of cobblestone dropped on the floor in the corner.
8 3 7 9
There's a chicken in the corner... how'd it get down here?
9 4 8 10
A room full of obsidian, how exciting...
10 1 5 11
You thought you heard a sound, but it's just your pet dog!
11 12 13 14
gjdhvijd
12 13 14 15
shjkfsehj
13 14 15 16
jkvsdhkjvk
14 15 16 17
jhfhjskkfc
15 16 17 18
hjksfhsakn
16 17 18 19
shfjshf
17 18 19 20
svjkwjfnsc
18 19 20 21
fjsjhv
19 20 21 22
vjksdhvujs
20 21 22 23
snvjkahf
21 22 23 24
svjjfk
22 23 24 25
snjkefhjw
23 24 25 1
snfjksjehlf
24 25 1 2
shfjkjneksf
25 1 2 3
hsjfpshje
```

```java
import java.util.*;

public class Rooms
{
    int roomCount;
    int currentRoom;
    int adjRoom1, adjRoom2, adjRoom3;
    String roomDescription;

    //constructor
    /** @param n the number of rooms
        @param c the current rooom
        @param a1 adjacent room 1
        @param a2 adjacent room 2
        @param a3 adjacent room 3
        @param d the room description **/
    public Rooms(int c,int a1,int a2,int a3, String d)
    {
        currentRoom=c;
        adjRoom1=a1;
        adjRoom2=a2;
        adjRoom3=a3;
        roomDescription=d;
    }

    public Rooms(Scanner info)
    {
        currentRoom=info.nextInt();
        adjRoom1=info.nextInt();
        adjRoom2=info.nextInt();
        adjRoom3=info.nextInt();
        roomDescription=info.nextLine();
        roomDescription=info.nextLine();
    }

    public void print()
    {
        System.out.println("You are in room "+currentRoom+"\n"+roomDescription+"\n"+
            "There are tunnels to rooms "+adjRoom1+", "+adjRoom2+", and"+adjRoom3+".");
    }
}
```

```
import java.io.*;
import java.util.*;

class HTW
{
    //global variables
    static int roomCount;
    static int currentRoom=1;
    static int adjRoom1;
    static int adjRoom2;
    static int adjRoom3;
    static Rooms [] cave;
    static int wumpus=1;
    static int arrows=3;
    static char action;
    static Scanner cin;
    static int play;
    static int wumpusRoom;
    static int spiderRoom;
    static int pitRoom;


    public static void main(String [] args) throws IOException
    {
        //reading in the text file of cave layout
        Scanner wumptext;
        wumptext=new Scanner(new FileReader("Wumpus.txt"));

        //allows person to type into program
        cin=new Scanner(System.in);

        roomCount=wumptext.nextInt();
        cave=new Rooms[roomCount];

        //picking trap rooms
        wumpusRoom=(int)(2+(roomCount-1)*Math.random());
        spiderRoom=(int)(2+(roomCount-1)*Math.random());
        pitRoom=(int)(2+(roomCount-1)*Math.random());

        //solving the problem if the trap rooms are the same
        if(spiderRoom==wumpusRoom || spiderRoom==pitRoom)
        {
            spiderRoom=(int)(2+(roomCount-1)*Math.random());
        }

        else if(pitRoom==wumpusRoom || pitRoom==spiderRoom)
        {
            pitRoom=(int)(2+(roomCount-1)*Math.random());
        }

        //loads room info into array
```

```java
    for(int i=0;i<cave.length;i++)
    {
      cave[i]=new Rooms(wumptext);
    }

    adjRoom1=cave[currentRoom-1].adjRoom1;
    adjRoom2=cave[currentRoom-1].adjRoom2;
    adjRoom3=cave[currentRoom-1].adjRoom3;

    //program intro
    System.out.println("Welcome to **Hunt The Wumpus!**\n");
    System.out.println("You are a mighty warrior, and armed with your trusty bow"
      +" and 3 arrows, you enter The Caves in search of the mighty Wumpus. If
you"
      +" shoot the Wumpus, you are victorious and the masses will praise you,
but"
      +" if you stumble upon the Wumpus unawares, it will eat you! Also, beware
of"
      +" the webs of the giant poisonous spiders and the bottomless pits!\n");
    System.out.println("Your senses of smell and hearing will aid you on your
quest,"
      +" for the Wumpus does not bathe and can be smelled one room away. Also,
the"
      +" clicking mandibles of the poisonous spiders can be heard one room
away,"
      +" and the foul odor of a bottomless pit can be smelled one room away.");

    play=Gameplay();
  }

  public static int Gameplay()
  {
    //loop makes game continue to run until the user runs out of arrows or
    wumpus dies
    do
    {
      //telling user the logistics of their locoation/stats
      System.out.println("\nYou are in room "+currentRoom+".");
      System.out.println("You have "+arrows+" arrows left.");
      System.out.println(cave[currentRoom-1].roomDescription);
      System.out.println("There are tunnels leading to rooms "
        +cave[currentRoom-1].adjRoom1+", "+cave[currentRoom-1].adjRoom2+", and
"
        +cave[currentRoom-1].adjRoom3+".");

      //asking user what their next move is
      System.out.println("Would you like to (M)ove or (S)hoot?");
      action=cin.next().charAt(0);

      //calls movement method if that's what the user chose to do
      if(action=='M' || action=='m')
```

```
            {
                int move=Movement();
                currentRoom=move;
            }

            //calls shooting method if that's what the user chose to do
            else if(action=='S' || action=='s')
            {
                int shoot=Shooting();
                if(shoot==1) {return 0;}
            }

            //prints error message if user enters anything else
            else
            {
                System.out.println("Oh no! Please enter either 'M' to move or"
                    +" 'S' to shoot!");
            }
        }while(wumpus==1 && arrows!=0);

        //if wumpus is dead (win)
        if(wumpus==0)
        {
            System.out.println("You shot the stinky Wumpus! *You win!**");
        }

        //if wumpus is alive and you have no arrows (loss)
        if(wumpus==1 && arrows==0)
        {
            System.out.println("You are out of arrows and so the smelly Wumpus lives"
                +" to see another day... **You lost the game.**");
        }

        return 0;
    }

    //movement
    public static int Movement()
    {
        System.out.println("Which room are you traveling to?");
        int choice=cin.nextInt();
        //System.out.println(choice!=cave[currentRoom-1].adjRoom1);
        //System.out.println(choice!=cave[currentRoom-1].adjRoom2);
        //System.out.println(choice!=cave[currentRoom-1].adjRoom3);


        //for invalid room choices
        if(choice!=cave[currentRoom-1].adjRoom1 && choice!=cave[currentRoom-1].adjRoom2
            && choice!=cave[currentRoom-1].adjRoom3)
        {
            System.out.println("Silly, that's not an option!");
            return currentRoom;
        }

        if(choice==cave[currentRoom-1].adjRoom1 ||
            choice==cave[currentRoom-1].adjRoom2 ||
            choice==cave[currentRoom-1].adjRoom3)
        {
            //close to wumpus warning
            if(cave[choice-1].adjRoom1==wumpusRoom ||
                cave[choice-1].adjRoom2==wumpusRoom ||
                cave[choice-1].adjRoom3==wumpusRoom)
            {
                System.out.println("You smell something stinky! The Wumpus is close.");
            }

            //close to spiders warning
            if(cave[choice-1].adjRoom1==spiderRoom ||
                cave[choice-1].adjRoom2==spiderRoom ||
                cave[choice-1].adjRoom3==spiderRoom)
            {
                System.out.println("You hear a faint clicking sound... spiders are 
near.");
            }

            //close to pit warning
            if(cave[choice-1].adjRoom1==pitRoom ||
                cave[choice-1].adjRoom2==pitRoom ||
                cave[choice-1].adjRoom3==pitRoom)
            {
                System.out.println("You hear screams echoing from a nearby pit.");
            }

            //wumpus death
            if(choice==wumpusRoom)
            {
                System.out.println("The stupid smelly Wumpus ate you... **You"
                    +" lost the game.**");
                System.exit(0);
            }

            //spider death
            else if(choice==spiderRoom)
            {
                System.out.println("Eek! You found the spiders! **You lost the 
game.**");
                System.exit(0);
            }

            //pit death
            else if(choice==pitRoom)
            {
                System.out.println("Yikes! You tumbled into the pit! **You lost the 
game.**");
                System.exit(0);
            }
            return choice;
        }
        return choice;
    }

    //shooting
    public static int Shooting()
    {
        System.out.println("Which room are you shooting your arrow into?");
        int choice=cin.nextInt();

        //for invalid room choices
        if(choice!=cave[currentRoom-1].adjRoom1 && choice!=cave[currentRoom-1].adjRoom2 
            && choice!=cave[currentRoom-1].adjRoom3)
        {
            System.out.println("You can't shoot there!");
            return arrows;
        }

        //correct choice
        if(choice==wumpusRoom)
        {
            wumpus=0;
            System.out.println("You shot the stinky Wumpus! **You win!**");
            return 1;
        }

        //incorrect choice
        else if(choice!=wumpusRoom)
        {
            System.out.println("Oof, you missed. The Wumpus is still alive and 
stinky.");
            arrows--;
        }
        return 0;
    }
}
```

```
import java.io.*;
import java.util.*;

class HTWec
{
    //global variables
    static int roomCount;
    static int spiderCount, pitCount;
    static int currentRoom=1;
    static int adjRoom1;
    static int adjRoom2;
    static int adjRoom3;
    static Rooms [] cave;
    static int wumpus=1;
    static int arrows=3;
    static char action;
    static Scanner cin;
    static int play;
    static int wumpusRoom;
    //static int spiderRoom;
    //static int pitRoom;
    static int supplyRoom;
    static int batRoom;
    public static int[] spiderRoom;
    public static int[] pitRoom;

    public static void main(String [] args) throws IOException
    {
        //reading in the text file of cave layout
        Scanner wumptext;
        wumptext=new Scanner(new FileReader("WumpusEC.txt"));

        //allows person to type into program
        cin=new Scanner(System.in);

        roomCount=wumptext.nextInt();
        cave=new Rooms[roomCount];

        spiderCount=wumptext.nextInt();
        spiderRoom=new int[spiderCount];
        pitCount=wumptext.nextInt();
        pitRoom=new int[pitCount];

        //picking trap rooms
        wumpusRoom=(int)(2+(roomCount-1)*Math.random());
        supplyRoom=(int)(2+(roomCount-1)*Math.random());
        while(supplyRoom==wumpusRoom)
        {
            supplyRoom=(int)(2+(roomCount-1)*Math.random());
        }
        for(int i=0;i<spiderCount;i++)
        {
            spiderRoom[i]=(int)(2+(roomCount-1)*Math.random());
            while(spiderRoom[i]==wumpusRoom || spiderRoom[i]==supplyRoom)
            {
                spiderRoom[i]=(int)(2+(roomCount-1)*Math.random());
            }
        }
        for(int i=0;i<pitCount;i++)
        {
            pitRoom[i]=(int)(2+(roomCount-1)*Math.random());
            while(pitRoom[i]==wumpusRoom || pitRoom[i]==spiderRoom[i] ||
pitRoom[i]==supplyRoom)
            {
                pitRoom[i]=(int)(2+(roomCount-1)*Math.random());
                for(int j=0;j<spiderCount;j++)
                {
                    if(pitRoom[i]==spiderRoom[j])
                    {
                        pitRoom[i]=(int)(2+(roomCount-1)*Math.random());
                    }
                }
            }
        }
        batRoom=(int)(2+(roomCount-1)*Math.random());

        //solving the problem if the trap rooms are the same
        /*if(spiderRoom==wumpusRoom || spiderRoom==pitRoom || spiderRoom==supplyRoom)
        {
            spiderRoom=(int)(2+(roomCount-1)*Math.random());
        }

        if(pitRoom==wumpusRoom || pitRoom==spiderRoom || pitRoom==supplyRoom)
        {
            pitRoom=(int)(2+(roomCount-1)*Math.random());
        }*/

        /* testing
        System.out.println("room count: "+roomCount);
        System.out.println("spider count: "+spiderCount);
        System.out.println("spider rooms: "+spiderRoom[0]);
        System.out.println("spider rooms: "+spiderRoom[1]);
        System.out.println("spider rooms: "+spiderRoom[2]);
        System.out.println("spider rooms: "+spiderRoom[3]);
        System.out.println("spider rooms: "+spiderRoom[4]);
        System.out.println("pit count: "+pitCount);
        System.out.println("pit rooms: "+pitRoom[0]);
        System.out.println("pit rooms: "+pitRoom[1]);
        System.out.println("pit rooms: "+pitRoom[2]);*/


        //loads room info into array
```

```
    for(int i=0;i<cave.length;i++)
    {
      cave[i]=new Rooms(wumptext);
    }

    adjRoom1=cave[currentRoom-1].adjRoom1;
    adjRoom2=cave[currentRoom-1].adjRoom2;
    adjRoom3=cave[currentRoom-1].adjRoom3;

    //program intro
    System.out.println("Welcome to **Hunting Herobrine!**\n");
    System.out.println("You are a mighty warrior, and armed with your trusty bow"
      +" and 3 arrows, you enter The Caves in search of the legend Herobine. If
you"
      +" shoot Herobrine, you are victorious and the masses will praise you,
but"
      +" if you stumble upon Herobine unaware, it will kill you! Also, beware
of"
      +" the webs of the poisonous spiders and the lava pits!\n");
    System.out.println("Your senses of smell and hearing will aid you on your
quest,"
      +" for Herobrine has a nametag and that can be seen one room away. Also,
the"
      +" clicking mandibles of the poisonous spiders can be heard one room
away,"
      +" and the crackling of the lava pit can be heard one room away.");

    play=Gameplay();
  }

  public static int Gameplay()
  {
    //loop makes game continue to run until the user runs out of arrows or
    wumpus dies
    do
    {
      //telling user the logistics of their locoation/stats
      System.out.println("\nYou are in room "+currentRoom+".");
      System.out.println("You have "+arrows+" arrows left.");
      System.out.println(cave[currentRoom-1].roomDescription);
      System.out.println("There are tunnels leading to rooms "
        +cave[currentRoom-1].adjRoom1+", "+cave[currentRoom-1].adjRoom2+", and
"
        +cave[currentRoom-1].adjRoom3+".");

      //asking user what their next move is
      System.out.println("Would you like to (M)ove or (S)hoot?");
      action=cin.next().charAt(0);

      //calls movement method if that's what the user chose to do
      if(action=='M' || action=='m')
```

```java
            {
                int move=Movement();
                currentRoom=move;
            }

            //calls shooting method if that's what the user chose to do
            else if(action=='S' || action=='s')
            {
                int shoot=Shooting();
                if(shoot==1) {return 0;}
            }

            //prints error message if user enters anything else
            else
            {
                System.out.println("Oh no! Please enter either 'M' to move or"
                    +" 'S' to shoot!");
            }
        }while(wumpus==1 && arrows!=0);

        //if wumpus is dead (win)
        if(wumpus==0)
        {
            System.out.println("You shot the creepy Herobrine! *You win!**");
        }

        /*//if wumpus is alive and you have no arrows (loss)
        if(wumpus==1 && arrows==0)
        {
            System.out.println("You are out of arrows and so the legend Herobrine
lives"
                +" to see another day... **You lost the game.**");
        }*/

        return 0;
    }

    //movement
    public static int Movement()
    {
        System.out.println("Which room are you traveling to?");
        int choice=cin.nextInt();

        //if you go in the bat room
        if(choice==batRoom)
        {
            System.out.println("Eek! A room full of bats! They're now going to pick"
                +" you up and drop you in a random room.");
            currentRoom=(int)(2+(roomCount-1)*Math.random());
            return currentRoom;
        }
        //for invalid room choices
        if(choice!=cave[currentRoom-1].adjRoom1 && choice!=cave[currentRoom-1].adjRoom2
            && choice!=cave[currentRoom-1].adjRoom3)
        {
            System.out.println("Silly, that's not an option!");
            return currentRoom;
        }

        if(choice==cave[currentRoom-1].adjRoom1 ||
            choice==cave[currentRoom-1].adjRoom2 ||
            choice==cave[currentRoom-1].adjRoom3)
        {
            //close to wumpus warning
            if(cave[choice-1].adjRoom1==wumpusRoom ||
                cave[choice-1].adjRoom2==wumpusRoom ||
                cave[choice-1].adjRoom3==wumpusRoom)
            {
                System.out.println("You see a nametag! Herobrine is close.");
            }

            //close to spiders warning
            for(int i=0; i<spiderCount; i++)
            {
                if(cave[choice-1].adjRoom1==spiderRoom[i] ||
                    cave[choice-1].adjRoom2==spiderRoom[i] ||
                    cave[choice-1].adjRoom3==spiderRoom[i])
                {
                    System.out.println("You hear a faint clicking sound... spiders are 
near.");
                }
            }

            //close to pit warning
            for(int i=0; i<pitCount; i++)
            {
                if(cave[choice-1].adjRoom1==pitRoom[i] ||
                    cave[choice-1].adjRoom2==pitRoom[i] ||
                    cave[choice-1].adjRoom3==pitRoom[i])
                {
                    System.out.println("You hear crackling from a nearby lava pit.");
                }
            }

            //wumpus death
            if(choice==wumpusRoom)
            {
                System.out.println("The stupid Herobrine killed you... **You lost the 
game.**");
                System.exit(0);
            }
            //spider death
            for(int i=0;i<spiderCount;i++)
            {
                if(choice==spiderRoom[i])
                {
                    System.out.println("Eek! You found the spiders! **You lost the 
game.**");
                    System.exit(0);
                }
            }

            //pit death
            for(int i=0;i<pitCount;i++)
            {
                if(choice==pitRoom[i])
                {
                    System.out.println("Yikes! You fell into the lava! **You lost the 
game.**");
                    System.exit(0);
                }
            }

            //supply room
            if(choice==supplyRoom)
            {
                System.out.println("Congrats, you found the arrow supply room!"
                    +" Your arrows will now be refilled back to 3!");
                arrows=3;
            }
            return choice;
        }
        return choice;
    }

    //shooting
    public static int Shooting()
    {
        System.out.println("Which room are you shooting your arrow into?");
        int choice=cin.nextInt();

        //for invalid room choices
        if(choice!=cave[currentRoom-1].adjRoom1 && choice!=cave[currentRoom-1].adjRoom2 
            && choice!=cave[currentRoom-1].adjRoom3)
        {
            System.out.println("You can't shoot there!");
            return arrows;
        }

        //correct choice
        if(choice==wumpusRoom)
        {
            wumpus=0;
            System.out.println("You shot Herobrine! **You win!**");
            return 1;
        }

        //incorrect choice
        else if(choice!=wumpusRoom)
        {
            System.out.println("Oof, you missed. Herobrine is still alive.");
            arrows--;
        }
        return 0;
    }
}
```