```java
public class MaxSeq {

 public static Node maxS (int[] table){
   int count = 0;
   Node[] nodes = new Node[table.length];

   Node n = new Node(table[0],1,1);

   for (int i = 1; i < table.length; i++){
     if (table[i] == n.getElem()) n.incLen();
     else {count = insertNode(nodes,n,count);
           n = new Node(table[i],1,1);}
   }
   count = insertNode(nodes,n,count);
   return maxNode(nodes, count);
 }

 public static int insertNode(Node[]ns, Node n, int count){
   boolean found = false;
   for (int i = 0; i < count && !found; i++){
      if (ns[i].getElem() == n.getElem() && ns[i].getLen() == n.getLen()){
            found = true;
            ns[i].incOccur();}
   }
   if (!found) {ns[count] = n; count++;}
   return count;
 }

 public static Node maxNode(Node[]ns, int count){
  display(ns, count);
  int n = 0;
  for (int i = 1; i < count; i++){
     if (ns[i].getLen() > ns[n].getLen()) n = i;
  }
  return ns[n];
 }

 public static void display(Node[] ns, int count){
   System.out.print("\n\nThe subsequences are:\n");
   for (int i = 0; i < count; i++){
     System.out.print("\n" + ns[i]);
   }
   System.out.print("\n");
 }

 /* public static int Nseqs (Node[] ns, int count, int L){
   int N = 0;
   for (int i = 0; i < count; i++)
      if (ns[i].getLen() == L) N = N + ns[i].getOccurs();
   return N;
 } */

 public static void main(String[] args){

  int[] table = new int[args.length];
  for (int i = 0; i < args.length; i++) table[i] = Integer.parseInt(args[i]);

  Node n = maxS(table);

  System.out.print("\n\nThus the first subsequence of max length is:\n" + n + "\n\n");


 }

}
```

```
public class Fstatistics {

public static void putchar (char c){System.out.print(c);}

public static void draw_line (int label, char form, int len){
  putchar('\n'); System.out.printf("%3d", label); putchar(' ');
  while (len > 0) {putchar(form); len--;}
}

public static void display_day_statistics(int day, int men, int women, int children){
  draw_line(day, 'M', (men + 5)/10);
  draw_line(day, 'W', (women + 5)/10);
  draw_line(day, 'C', (children + 5)/10);
  draw_line(day, 'T', (men+women+children+5)/10);
}

public static void main(String[] args){
 int Day = 1, Men, Women, Children;
 int MaxM = 0, MaxW = 0, MaxC = 0, MaxT = 0, DM = 0, DW = 0, DC = 0, DT = 0;
 while(!StdIn.isEmpty()){
   Men = StdIn.readInt();
   Women = StdIn.readInt();
   Children = StdIn.readInt();
   display_day_statistics(Day, Men, Women, Children);
   if (Men > MaxM) {MaxM = Men; DM = Day;}
   if (Women > MaxW) {MaxW = Women; DW = Day;}
   if (Children > MaxC) {MaxC = Children; DC = Day;}
   if (Men + Women + Children > MaxT) {MaxT = Men + Women + Children; DT = Day;}
   Day++;}
 System.out.printf("\n\nMax no of men was %4d on day %3d", MaxM, DM);
 System.out.printf("\nMax no of women was %4d on day %3d", MaxW, DW);
 System.out.printf("\nMax no of children was %4d on day %3d", MaxC, DC);
 System.out.printf("\nMax no of total passengers was %4d on day %3d\n\n", MaxT, DT);
 }
}


public class Caps {

  public static char capL (char c) {
     if ('a' <= c && c <= 'z')
     return (char) ((int) c - (int) 'a' + (int) 'A');
     else return c;
  }

  public static void main (String [] args) {
    while (!StdIn.isEmpty()) {
       char c = StdIn.readChar();
       System.out.print(capL(c));
    }
  }

}


public class LineN{

 public static void main (String[] args){
    int count=1; System.out.printf("%3d: ", count);
    while (!StdIn.isEmpty()){
              char c = StdIn.readChar();
              System.out.print(c);
              if (c=='\n') {count++; System.out.printf("%3d: ", count);}

    }
  }

}
```

```
public class WCount {

      public static boolean white_space (char c) {
        return c == ' ' || c == '\n' || c == '\t';
      }

      public static void main (String[] args) {
        char c = ' ';
          boolean wflag = false;
        int words=0, lines=0, chars=0;
        while(!StdIn.isEmpty()) {
                 c = StdIn.readChar();
            chars++;
            if (c == '\n') lines++;
            if (white_space(c) && wflag) wflag = false;
            else if (!white_space(c) && !wflag) {words++; wflag=true;}
        }
          if (c != '\n') lines++;
        System.out.printf("chars = %d, words = %d, lines = %d", chars, words, lines);
      }
}
```

```
public class XmasFigs {

public static void snowManRow (int row){
 switch (row){
  case 1:  System.out.printf("      _______      "); break;
  case 2:  System.out.printf("     /       \\     "); break;
  case 3:  System.out.printf("     | x   x |     "); break;
  case 4:  System.out.printf("     |   O   |     "); break;
  case 5:  System.out.printf(" _|   \\ --- /  |_  "); break;
  case 6:  System.out.printf("   \\  -------  /   "); break;
  case 7:  System.out.printf("    \\/       \\/    "); break;
  case 8:  System.out.printf("     |   o   |     "); break;
  case 9:  System.out.printf("     |       |     "); break;
  case 10: System.out.printf("     |   o   |     "); break;
  case 11: System.out.printf("     |       |     "); break;
  case 12: System.out.printf("     \\_______/     "); break;
  case 13: System.out.printf("   * MERRY XMAS *  ");
}}

public static void starRow (int row){
 switch (row){
   case 1: System.out.print("     *     "); break;
   case 2: System.out.print("    ***    "); break;
   case 3: System.out.print(" ********* "); break;
   case 4: System.out.print("  *******  "); break;
   case 5: System.out.print(" ********* "); break;
   case 6: System.out.print("    ***    "); break;
   case 7: System.out.print("     *     ");
}}

public static void xmasRow (int row){
 switch (row){
   case 1: System.out.print("           "); break;
   case 2: System.out.print("           "); break;
   case 3: System.out.print("   MERRY   "); break;
   case 4: System.out.print("  X M A S  "); break;
   case 5: System.out.print(" ********* "); break;
   case 6: System.out.print("           "); break;
   case 7: System.out.print("           ");
}}

public static void blankRow(){
  System.out.print("           ");
}

public static boolean[][] instantiate (int size){
  boolean[][] t = new boolean[size][size];
  for (int i = 0; i < size; i++)
    for (int j = 0; j < size; j++)
       t[i][j] = StdIn.readBoolean();
  return t;
}

public static void drawStars(boolean[] bs, boolean mid){
  int count = bs.length/2;
  for (int i = 1; i <= 7; i++){
       System.out.println();
       for (int j = 0; j < bs.length; j++)
          if (mid && j == count) xmasRow(i);
          else if (bs[j]) starRow(i);
          else blankRow();
  }
}

public static void main (String[] args){
 int choice = Integer.parseInt(args[0]);
 if (choice == 1){
     int count = Integer.parseInt(args[1]);
     for (int i = 1; i <=13; i++){
       System.out.println();
       for (int j = 1; j <= count; j++) snowManRow(i);
     }
     System.out.println();
  }
 else {int size = StdIn.readInt();
       int mid = size / 2;
       boolean[][] blanks = instantiate(size);
       for (int row = 0; row < blanks.length; row++){
           if (row == mid) drawStars(blanks[row], true);
           else drawStars(blanks[row], false);
         }
       }
}

}

public class CreatePattern {

 public static void main (String[] args){
  int size = Integer.parseInt(args[0]);
  System.out.print(size + "\n");
  boolean[][]t = new boolean[size][size];
  for (int i = 0; i < size; i++) t[i][i] = true;
  for (int i = size-1; i >= 0; i--) t[i][size-1-i] = true;
  for (int i = 0; i < size; i++){
   for (int j = 0; j < size; j++)
      if(t[i][j]) System.out.print("true\t");
      else System.out.print("false\t");
   System.out.print("\n");
  }
 }
}
```