



THE UNIVERSITY  
OF AUCKLAND

# COMPSCI 230

Software Design and Construction

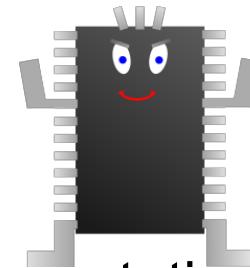
Game Example

2013-05-13

# Swing and Threads



# Threads



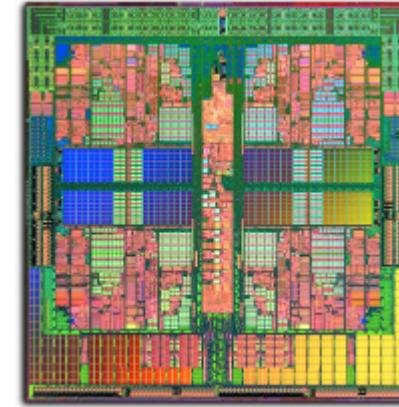
**Thread** of program execution: computational resource that executes instructions of a program, usually consisting of...

- **Program counter**: pointer to next instruction to be executed
- **Call stack** with local variables and return addresses
- Set of **registers** for storing data directly in the processor



## Multi-threaded program

- A program that uses multiple threads to execute several portions of code concurrently
- Each thread can potentially run on a different CPU / core



Quad-Core AMD  
Opteron

# SWING AND THREADS



- One event-dispatching thread
- All event handlers run by event-dispatching thread
- **Single-thread rule:** all code accessing a widget (after creating it) should be executed in the event-dispatching thread
  - Swing is not thread-safe
  - Only a few widget methods that are thread-safe
- **Responsiveness rule:** don't do time-consuming tasks in event listeners; create new threads instead

**Responsiveness  
Rule**



**Single-Thread Rule**

# Using Worker Threads for Responsiveness

```
...
JButton button = new JButton("Invoke");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Thread t = new Thread(new Runnable() {
            public void run() {
                // Do some long operation...
                // (computation or I/O)
                // Change model; view will follow...
            }
        });
        t.start();
    }
});
...
...
```

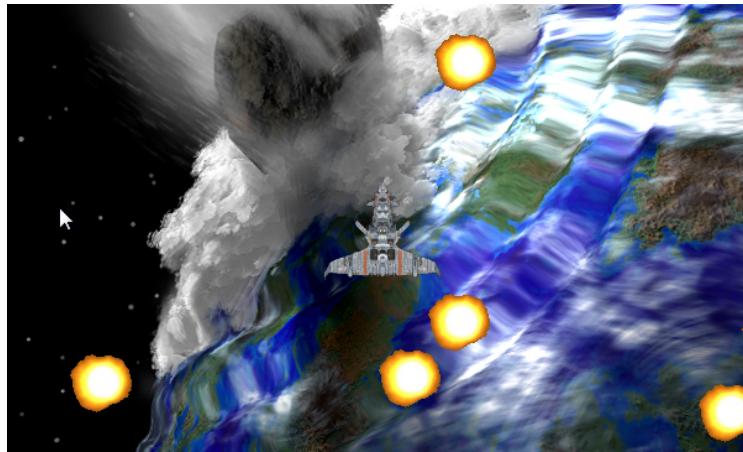
# Doing GUI Changes on the Event Dispatching Thread

```
...
Thread t = new Thread(new Runnable() {
    public void run() {
        // Cannot access GUI directly from here (unsafe).
        // Let the event dispatching thread do it:
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                // access the GUI here
            }
        });
    }
});
t.start();
...
```

**Alternative:** `invokeAndWait()` waits until the event dispatching thread has executed the `run()` method

# Game Example

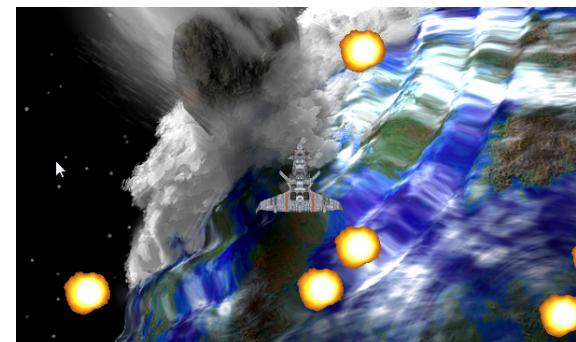
## MeteorField



# MeteorField Part 1

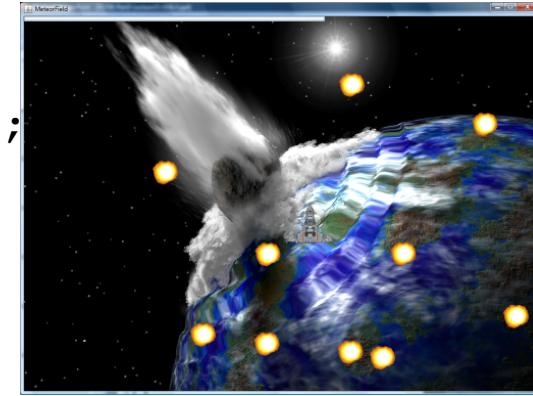
## Main Class

```
public class MeteorField extends JFrame {  
    final static int numOfMeteors = 10;  
    static ImageIcon image = new ImageIcon("background.jpg");  
    public static int width = image.getIconWidth();  
    public static int height = image.getIconHeight();  
    public static Hero hero;  
    public static JLabel hitLabel; int hits = 0;  
    public static Meteor[] meteors;  
    public static boolean up = false, down = false,  
                    left = false, right = false;  
  
    public MeteorField() {  
        // see next slides...  
    }  
  
    public static void main(String[] args) {  
        // see next slides...  
    }  
}
```



# MeteorField Part 2: GUI Setup

```
public MeteorField() {  
    setTitle("MeteorField"); setSize(width, height);  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    getContentPane().setLayout(null);  
    JLayeredPane pane = new JLayeredPane();  
    pane.setBounds(0,0, width, height);  
    JLabel background = new JLabel(image);  
    background.setBounds(0,0, width, height);  
    pane.add(background, new Integer(1));  
    meteors = new Meteor[numOfMeteors];  
    for(int i =0; i<numOfMeteors; i++) {  
        meteors[i] = new Meteor();  
        pane.add(meteors[i], new Integer(3));  
    }  
    hero = new Hero(); pane.add(hero, new Integer(2));  
    hitLabel = new JLabel("0");  
    hitLabel.setBounds(10,10,50,50);  
    pane.add(hitLabel, new Integer(4));  
    getContentPane().add(pane); setVisible(true);  
    // continued in part 3...
```



# MeteorField Part 3

## Event Handler Setup

```
addKeyListener(new KeyAdapter() {
    public void keyPressed(KeyEvent evt) {
        if(evt.getKeyCode() == KeyEvent.VK_UP) {
            up = true; down = false;
        }
        if (evt.getKeyCode() == KeyEvent.VK_DOWN) {
            down = true; up = false;
        }
        if (evt.getKeyCode() == KeyEvent.VK_LEFT) {
            left = true; right = false;
        }
        if (evt.getKeyCode() == KeyEvent.VK_RIGHT) {
            right = true; left = false;
        }
    }

    public void keyReleased(KeyEvent evt) {
        if(evt.getKeyCode() == KeyEvent.VK_UP) up = false;
        if(evt.getKeyCode() == KeyEvent.VK_DOWN) down = false;
        if(evt.getKeyCode() == KeyEvent.VK_LEFT) left = false;
        if(evt.getKeyCode() == KeyEvent.VK_RIGHT) right = false;
    }
});
```



# MeteorField Part 4

## Main Method

```
public static void main(String[] args) {  
    final Frame frame = new MeteorField();  
    java.util.Timer timer = new java.util.Timer();  
    timer.schedule(new TimerTask() {  
        public void run() {  
            hero.move();  
            Rectangle heroRect = new Rectangle(  
                hero.getX(), hero.getY(), Hero.width, Hero.height);  
            for(int i=0; i<numOfMeteors; i++) {  
                Meteor m = meteors[i]; m.move();  
                Rectangle meteorRect = new Rectangle(  
                    m.getX(), m.getY(), width, height);  
                if(heroRect.intersects(meteorRect)) {  
                    hits++; hitLabel.setText(" " + hits); }  
                }  
                if(hits>1000) {  
                    JOptionPane.showMessageDialog(frame , "Game Over!");  
                    System.exit(0);  
                }  
            }  
        }, 0, 5); }
```



# MeteorField Part 5

## The Hero

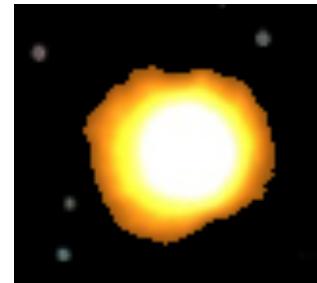
```
class Hero extends JLabel {  
    static ImageIcon image = new ImageIcon("hero.png");  
    public static int width = image.getIconWidth();  
    public static int height = image.getIconHeight();  
  
    public Hero() {  
        super(image);  
        setSize(width, height);  
        setLocation(MeteorField.width/2, MeteorField.height/2);  
    }  
  
    void move() {  
        if (MeteorField.up && getY()>0)  
            setLocation(getX(), getY()-1);  
        if (MeteorField.left && getX()>0)  
            setLocation(getX()-1, getY());  
        if (MeteorField.down && getY()+height<MeteorField.height)  
            setLocation(getX(), getY()+1);  
        if (MeteorField.right && getX()+width<MeteorField.width)  
            setLocation(getX()+1, getY());  
    }  
}
```



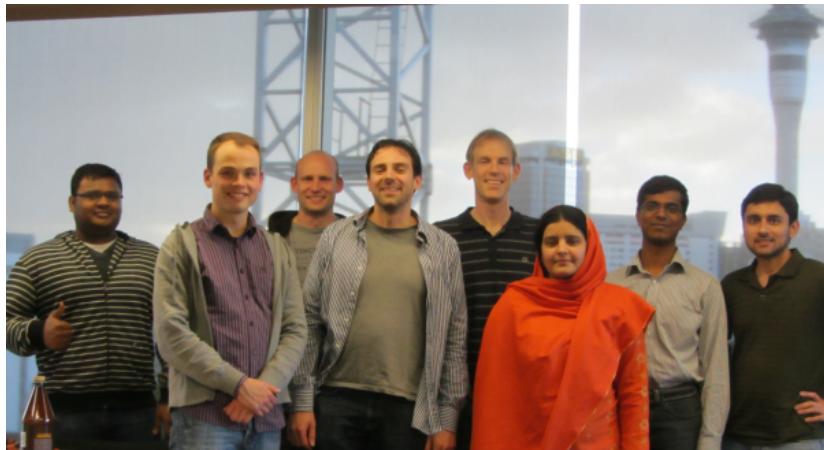
# MeteorField Part 6

## The Meteors

```
class Meteor extends JLabel {  
    // define image, width and height analogously to Hero...  
    static Random rand = new Random();  
    float x, y, dx, dy;  
  
    public Meteor() {  
        super(image);    setSize(width, height);  
        x = rand.nextInt(800 - Meteor.width);  
        y = rand.nextInt(600 - Meteor.height);  
        dx = (rand.nextInt(2)==0)?  
            (0.5f+rand.nextFloat()/2):(-0.5f-rand.nextFloat()/2);  
        dy = (rand.nextInt(2)==0)?  
            (0.5f+rand.nextFloat()/2):(-0.5f-rand.nextFloat()/2);  
    }  
  
    void move() {  
        x += dx;    y += dy;  
        if(x<=0) dx = -dx;    if(y<=0) dy = -dy;  
        if(x + width >= MeteorField.width) dx = -dx;  
        if(y + height >= MeteorField.height) dy = -dy;  
        setLocation((int)x, (int)y);  
    } }
```

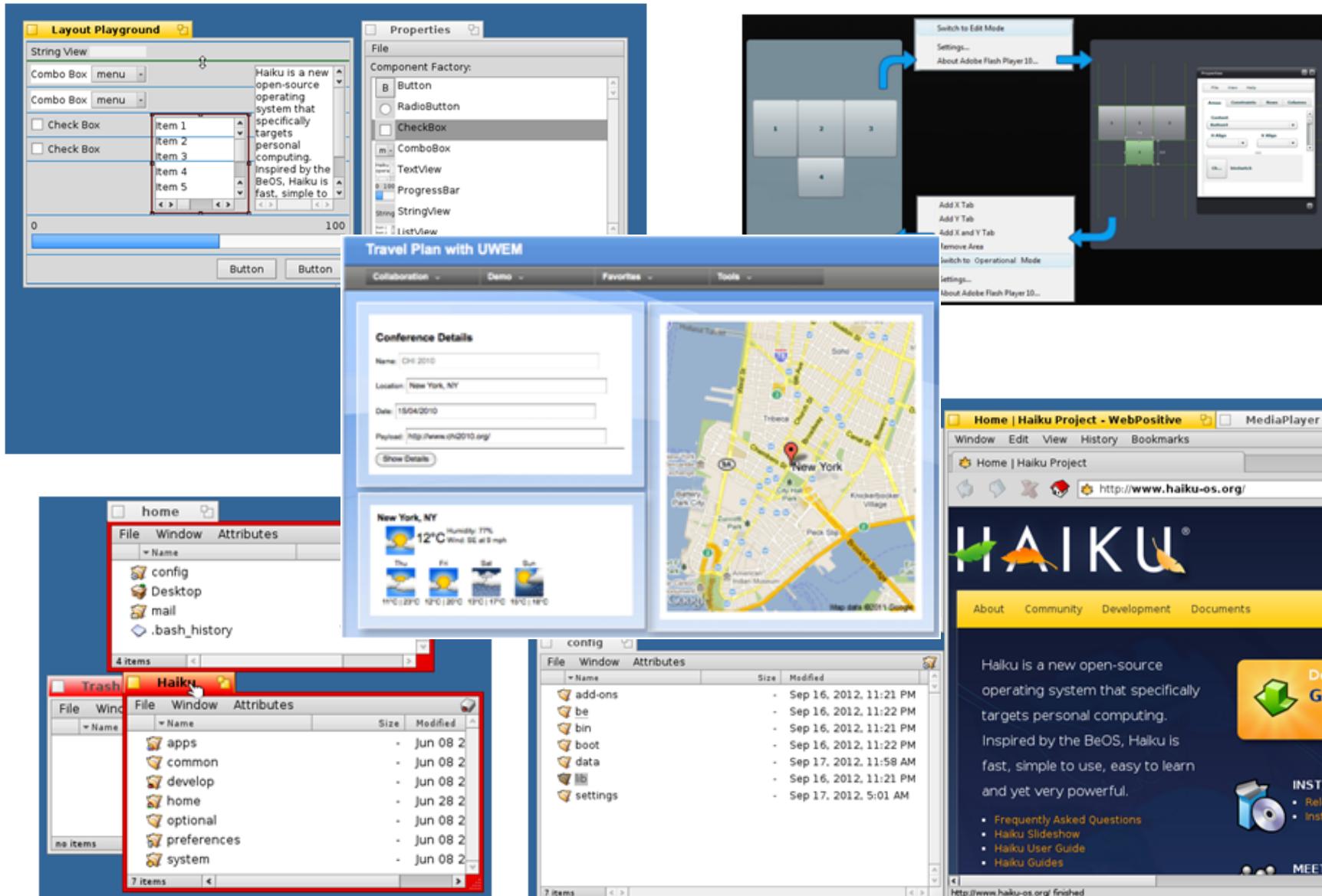


# User Interface Research

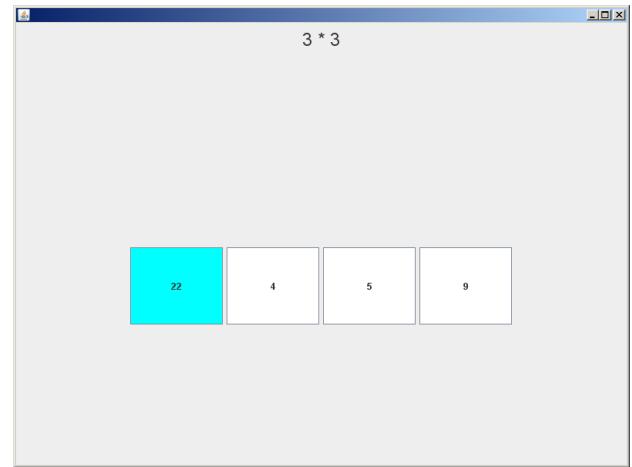
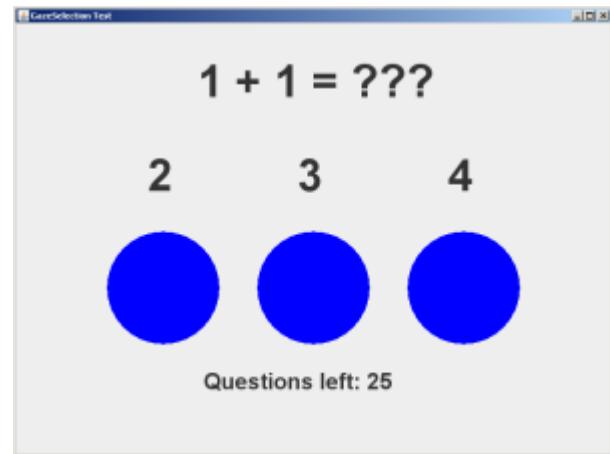


*If we knew what it was we were doing, it would not be called research, would it?  
(Albert Einstein)*

# User Interface Technology



# Eye Tracking



```

#include <math.h>
#include <sys/time.h>
#include <X11/Xlib.h>
#include <X11/Xkeysym.h>
double L ,o=1,d,
      _wdt,T,Z,D=1,d,
      s[999],E,h=8,I,
      J,K,w(999),M,O
      ,n(999),j=33e-3,i=
      1E3,x,t,u,v,W,S=
      74.5,1<221,Xe7.26,
      a,B=A=32.2,c,F,H;
      int N,q,C,Y,p,U;
      Window z; char f[52];
      GC k; main() {Display=e=
      XOpenDisplay(0); z=RootWindow(e,0); for (XSetForeground(e,k=XCreateGC (e,z,0,0),BlackPixel(e,0))
      ; scanf("%lf%lf%lf",y+=w,y,y+=z)+1; y++); XSelectInput(e,z= XCreateSimpleWindow(e,z,0,400,400,
      0,0,WhitePixel(e,0)),KeyPressMask); for (XMapWindow(e,z); T=sin(O) { struct timeval G(0,dt*1e6
      ); K= cos(j); N=le4; M= H_= Z=D*K; F=_*p; x=E*K; W=cos(O); H=K*T; O+=D_*F/ K+d/K*E_= B=
      sin(j); a=B*T*D-E*W; XCLEARWindow(e,z); t+=E*D*B*W; j+=d_-D_-*F*E; F+=W*E*B-T*D; for (o+=I*D+W+E
      *T*B,E/d/K*B+v+B*K*F*D)*; pcy: ) { T=p[s]+1; E=d[p]-L; K=D*m-B*T-H*E; if(p [n]+w(p[p]+s
      )== 0|K <fabs(W*T*x-I*E+D*p) |fabs(D*t*D*x-T*E) > K*N=le4: else{ q=W*H *482+2e2; C= 2E2+4e2/ K
      *D; N=1E466 XDrawLine(e,z,k,N,u,q,O); U=c; l+=_*( X*t +B*M*m+1); T-X*X+ 1*I+M *M;
      XDrawString(e,z,k ,20,380,f,17); Dv=1/I*5; i=(B-1*M*x-XZ)^2; for (XPending(e); u *=CS!=N){ XEvent
      XNextEvent(e,&ev);
      +*(XLookupKeySym
      (ev.keysym,0))-177
      N-L?7 UP-N?6 E:6
      J,g u: sh: --*(
      Dm-N- D? ?N=-
      RT?6u: & W:sh:6J
      ); } m=15*F/1;
      c+= (I*W/ 1,1*H
      +I*M+a*X)*.; H
      +A*x+v*X-F*1+
      E=.1*X4.9/1.t
      =T*m/32-I*T/24
      )/S; K=F*W*(
      h* le4/1-(T+
      E*5*T*E)/3e2
      )/S-X*d-/B*A;
      a=2.63 /1*d;
      X=( d*I-T/S
      *(.19*+a
      *.64+j/1e3
      )-M* v +*a
      )Z=; l +=_
      K; _w: W;d;
      sprint(f,
      "5d
      "97d", p+=1
      /1.7, (C-93+
      O*57.3)%050,
      (int)i: d+=T* (.45-14/1*
      X-a*130-J*-.14)*. /125*a*F*-t; p=(T*(47
      *I-* 52*E*94 +D*t-.38*u+.21*E) /1e2+W*
      179*v)/2312; select(p=0,0,0,.6G); v=-
      W*F-T*( .63*m-I*.086+m*E*19-D*25-.11*u
      )/107e2); _w: D*cos(o); E=sin(o); } }
      }

      /*-----*/
      #include <time.h>
      #define c(C)/* - . */return ( C); /* 2004*/
      #include <stdio.h>/. Meekan
      *typedef/* *char prp* u
      *l[128],*v*typedef int _J_ R,i,N,I,A
      ,m,o,e
      [9], a[256],k [9], n[ 256]:FILE*f
      ,r_x(_K,_z
      ,_q): for(i
      ,q=0x0000ffff, l=(K>>8)) {
      _u[0] = E
      open( (p+r, prq ))( c(
      r_++ , q ))=c(
      ,q=q) _D(c( fclose(f) ), B(_q)(c( fseek
      (f, 0
      ,q), -1,0))); B(_q)(c( puts(q
      ,q) )); /* /
      /main( t,p**x)(if((q)c( C(<ln
      ,file>)*"file"> )) u=0;i=I*(E([1],"zb")) 2B(2)20 : ((o =ftell
      (f))>8)?(o=(p*)malloc(o):B(0)?0:fread(u,o,1,f):0):0:0: D():0;
      (u)c("bad40input "));if(E([2],"zb"))for(N=1;256<n[i++]-1=i[
      i]=0;
      for(1=0; i<6&R=fgetcl( f )>-1;i++)a[R]?G=H? 2->D? )?n[i]
      :1)?0:(R[1]=164N&a[1]):N=0; [1]=R-1; i=1; i<27; i++
      ) n[i]=1+164N&a[1]? N=a [1]:0; i>(im=I0):15(A-1) for(Nen[A];
      I< 864 (R=getccl(f))> -164 <0
      : mu [1]?1?((u[I]= -46)?(a
      , "bad40input " "you" )for(i
      , A >0): ( (R4)=0) ?(unsigned int)R>>(01): (unsigned
      /*kerro Q' ,MS "/*)>> 2) / Ouedba88320;numma[I-1];i;
      j=m <N? (mm N=8): ++ mfor(m=0;1<I;e[i+1]=0)(
      vvw [1]+1;for(R
      R=(_)* W[i]* (
      R+= G*/ (*u+,
      "%*n" "Ex," x=Retime(i=0,m,o)-
      0) ; i< 8++ , 1)u [N-1]="" (*u+i+1):for(*k*x(-
      0,1*0 , *a):i>= 1) for (A=i;A<I+A+)(u[+][ A ]
      )=>(A [e[A]]: k [A+1]= (K[A],a[A],s[A+1]
      ));if( R==k[I]) o (E([2]),"ubw" )?fwrite(
      ()|C("In OK."):0 :c(
      "\n WriteError"
      , "\n WriteError"
      ) for ( i =+I-
      1; i>-1?i+1)[+]
      , for( A=i+;
      =>0); ( i <I-4
      ( _ ) 46
      /*)
      ); o
      (" \n
      /*)
      dP
      , pd '
      , zo
      */
      }

      International Obfuscated C Code Contest (ioccc.org)
  
```

# Thank you!