

AMD Programming Contest Results

Hosted by the IEEE Computer Society

March 28, 2008

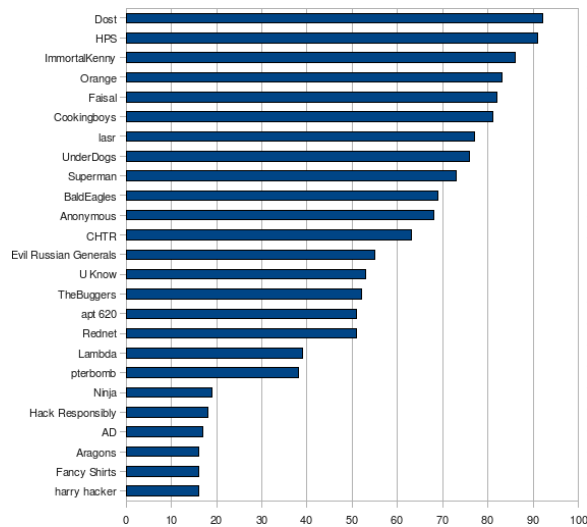
1 Contest Results

Congratulations to the winners of the AMD Programming contest. We thoroughly enjoyed hosting the event and were excited to see interest from both ECE and Computer Science majors. Ten teams solved all five puzzles, so the winners were determined by the optimizations you made for the extra grading criteria. The grade distribution and some of the best submissions appear below. We hope to see you all competing again next year!

1st Place: Team Dost – Gaurav Nolkha, Birgi Tamersoy

2nd Place: Team HPS – Rustam Miftakhutdinov, Veynu Narasiman

3rd Place: Team ImmortalKey – Richard Naething, Sterling Wei



2 Bitbin

Author: Derrick Huhn

Winners: Evil Russian Generals – Nicu Sturca, Chad LaGuardia

Solution size: 506 bytes

```
from math import *
def f(n):
    if n<2:
        return 1
```

```

y=1
while n:
    y*=n
    n-=1
return y
def O(x):
    y=0
    while x:
        if x%2:
            y+=1
        x/=2
    return y
def K(n,k):
    if k>n:
        return 0
    y=1
    t=f(k)
    while k:
        y*=n
        n-=1
        k-=1
    y/=t
    return y
def c(p,m):
    if p==0:
        return 0
    if m==0:
        return -1
    n=int(log(m)/log(2))
    return K(n,p)+c(p-1,m-2**n)
s=input()
e=input()
z=input()
y=0
P=[2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61]
for p in P:
    if z%p==0:
        y+=c(p,e)-c(p,s)
        if O(s)==p:
            y+=1
print y

```

3 Break This Protocol

Author: Derrick Huhn

Winners: Faisal - Faisal Iqbal and Junaid

Submission time: 03/19/2008 23:17:20

1. charlie used {q}Ka from SAMPLE_EXECUTION and gave it as {Z}Ka to the server in a new session
2. server responded with {q}Kc and charlie found q
3. charlie takes {Kab}q from SAMPLE_EXECUTION and finds Kab and then decrypts SECRET

The given source was used with hardcoded values to find the secret message and the server was invoked for the above message transaction

Answer:

Once you find the cheese, don't forget to do the binomial dance

4 Maze

Author: Patrick Lowry

Winners: Lambda - Sean Catchpole and Gandiinaa Gumenjav

```

#include <stdio.h>
#include <stdlib.h>

```

```

#define NORTH (char)(1<<7)
#define SOUTH (char)(1<<6)
#define EAST (char)(1<<5)
#define WEST (char)(1<<4)
#define UP (char)(1<<3)
#define DOWN (char)(1<<2)
#define RIGHT (char)(1<<1)
#define LEFT (char)(1)

#define COUNT(m) ((m)>>7&1) + ((m)>>6&1) + ((m)>>5&1) + ((m)>>4&1) + ((m)>>3&1) + ((m)>>2&1) + ((m)>>1&1) + ((m)&1);

int main(int argc, char* argv[] ) {

    if(argc!=2) { fprintf(stderr,"usage: %s file",argv[0]); exit(1); }

    // File I/O
    FILE * mazefile;
    long mazesize;
    char * maze;
    size_t result;

    mazefile = fopen( argv[1], "rb" );
    if(mazefile==NULL) { fputs("File error",stderr); exit(2); }

    // obtain file size:
    fseek(mazefile, 0, SEEK_END);
    mazesize = ftell(mazefile);
    rewind(mazefile);

    // allocate memory to contain the whole file:
    maze = (char*) malloc(sizeof(char)*mazesize);
    if(maze == NULL) { fputs("Memory error",stderr); exit(3); }

    // copy the file into the buffer:
    result = fread (maze,1,mazesize,mazefile);
    if(result != mazesize) { fputs("Reading error",stderr); exit(4); }

    // terminate
    fclose (mazefile);

    // Maze Size
    int size = 0;
    int end = mazesize-1;
    while(size*size*size*size<mazesize) ++size;
    //printf("Size: %d\n",size);

    // Solve Maze
    char m,p;
    int i,k,n,z;
    int a,b,c,d;
    int cc=size;
    int bb=cc*size;
    int aa=bb*size;
    int again;

    do {
        again=0;
        for(i=end; i-->1;) {
            m = maze[i];
            n = COUNT(m);
            // Fill Dead End
            if(n==1) {
                z = i;
                again=1;
                a=(z/aa)%size;
                b=(z/bb)%size;
                c=(z/cc)%size;
                d=z%size;
                //printf("Dead End a[%d] b[%d] c[%d] d[%d] z[%d]\n",a,b,c,d,z);
                while(n==1) {
                    maze[z]=0;
                    switch(m) {
                        case NORTH: --a; m=SOUTH; break;
                        case SOUTH: ++a; m=NORTH; break;
                        case EAST: --b; m=WEST; break;
                        case WEST: ++b; m=EAST; break;
                    }
                }
            }
        }
    } while(again);
}

```

```

        case UP:    --c; m=DOWN; break;
        case DOWN: ++c; m=UP;   break;
        case RIGHT: --d; m=LEFT; break;
        case LEFT:  ++d; m=RIGHT; break;
    }
    z = a*aa + b*bb + c*cc + d;
    maze[z] &= ~m;
    m = maze[z];
    n = COUNT(m);
    //printf("Adjusted a[%d] b[%d] c[%d] d[%d] z[%d]\n",a,b,c,d,z);
    if(!z || z==mazesize-1) break; //dont remove ends
    }
    //printf("\n");
}
} while(again);

/*
// Output Readable maze
int num[9] = {0,0,0,0,0,0,0,0,0};
for(i=mazesize; i-->0;) {
    m = maze[i];
    n = COUNT(m);
    ++num[n];
}

for(i=0;i<9;i++) {
    printf("num[%d] = %d\n",i,num[i]);
}
*/

// Output solution
a=b=c=d=z=m=0;
while(z!=end){
    m = maze[z] & ~m;
    switch(m) {
        case NORTH: --a; m=SOUTH; p='N'; break;
        case SOUTH: ++a; m=NORTH; p='S'; break;
        case EAST:  --b; m=WEST;  p='E'; break;
        case WEST:  ++b; m=EAST;   p='W'; break;
        case UP:    --c; m=DOWN;   p='U'; break;
        case DOWN:  ++c; m=UP;     p='D'; break;
        case RIGHT: --d; m=LEFT;   p='R'; break;
        case LEFT:  ++d; m=RIGHT;  p='L'; break;
    }
    printf("%c",p);
    z = a*aa + b*bb + c*cc + d;
}

free (maze);
return 0;
}

```

5 Zombie Administrators

Author: Derrick Huhn

Winners: Superman - Cameron Davison and Chris Craik

Solution size: 44 bytes

```
while(($i<>)-'q'){printf("%.0f\n",2**$i/4)}
```

6 CPU Simulator

Author: Doug Reed and Patrick Lowry

Winners: Dost - Gaurav Nolkha and Birgi Tamersoy

Source available at http://ieeecs.ece.utexas.edu/challenges/dost_sim.c.