

Zadanie 3: Pomôcka na testovanie

```
int main()
{
    char **mapa;
    int i, test, dlzka_cesty, cas, *cesta;
    int n=0, m=0, t=0;
    FILE* f;
    while(1){
        printf("Zadajte cislo testu (0 ukonci program):\n");
        scanf("%d",&test);
        dlzka_cesty = 0;
        n=m=t=0;
        switch(test){
            case 0://ukonci program
                return 0;
            case 1://nacitanie mapy zo suboru
                f=fopen("test.txt","r");
                if(f)
                    fscanf(f, "%d %d %d", &n, &m, &t);
                else
                    continue;
                mapa = (char**)malloc(n*sizeof(char*));
                for(i=0; i<n; i++){
                    mapa[i] = (char*)malloc(m*sizeof(char));
                    for (int j=0; j<m; j++){
                        char policko = fgetc(f);
                        if(policko == '\n') policko = fgetc(f);
                        mapa[i][j] = policko;
                    }
                }
                fclose(f);
                cesta = zachran_princezne(mapa, n, m, t, &dlzka_cesty);
                break;
            case 2://nacitanie preddefinovanej mapy
                n = 10;
                m = 10;
                t = 12;
                mapa = (char**)malloc(n*sizeof(char*));
                mapa[0]="CCHCNHCCHN";
                mapa[1]="NNCCCHHCCC";
                mapa[2]="DNCCNNHHHC";
                mapa[3]="CHHHCCCCC";
                mapa[4]="CCCCCNHHHH";
                mapa[5]="PCHCCNNNN";
                mapa[6]="NNNNNHCCCC";
                mapa[7]="CCCCCPCCCC";
                mapa[8]="CCCNHHHHH";
                mapa[9]="HHHPCCCCC";
                cesta = zachran_princezne(mapa, n, m, t, &dlzka_cesty);
                break;
            case 3: //pridajte vlastne testovacie vzorky
                default:
                    continue;
        }
        cas = 0;
        for(i=0; i<dlzka_cesty; i++){
            printf("%d %d\n", cesta[i*2], cesta[i*2+1]);
            if(mapa[cesta[i*2+1]][cesta[i*2]] == 'H')
                cas+=2;
            else
                cas+=1;
            if(mapa[cesta[i*2+1]][cesta[i*2]] == 'D' && cas > t)
                printf("Nestihol si zabit draka!\n");
            if(mapa[cesta[i*2+1]][cesta[i*2]] == 'N')
                printf("Nestihol si zabit draka!\n");
        }
    }
}
```

```

        printf("Prechod cez nepriechodnu prekazku!\n");
        if(i>0 && abs(cesta[i*2+1]-cesta[(i-1)*2+1])+abs(cesta[i*2]-cesta[(i-1)*2])>1)
            printf("Neplatny posun Popolvara!\n");
    }
    printf("%d\n",cas);
    free(cesta);
    for(i=0; i<n; i++){
        free(mapa[i]);
    }
    free(mapa);
}
return 0;
}

```

Vstup zo súboru:

```

10 10 12
CCHCNHCCHN
NNCCCHHCCC
DNCCNNHHHC
CHHHCCCCC
CCCCNHHHH
PCHCCCNNN
NNNNNHCCCC
CCCCPCCCC
CCCNHHHHH
HHHPCCCCC

```

Najkratšia cesta s časom 29:

```

0 0
1 0
2 0
2 1
2 2
2 3
1 3
0 3
0 2
0 3
0 4
0 5
1 5
2 5
3 5
4 5
5 5
5 6
5 7
5 8
5 9
4 9
3 9

```

Výstup testu je tiež možné vizualizovať v nástroji <https://popolvar.surge.sh/>.

