

7. Српска информатичка олимпијада
Београд – 18. мај 2013 .

1.

```
#include <stdio.h>
int main()
{
    int N, M, max, i, j, k;
    scanf("%d%d", &N, &M);
    int zad[N][M];

    for(i = 0; i < N; i++)
        for(j = 0; j < M; j++) scanf("%d", &zad[i][j]);

    int suma[M], suma_kol[N];
    for(i = 0; i < N; i++) suma_kol[i] = 0;
    for(i = 0; i < M; i++) suma[i] = 0;
    int maxi = 0, maxj = 0, max_kol = 0;

    for(k = 0; k < M; k++)
    {
        max = 0;
        int b;
        for(b = 0; b < N; b++)
            suma_kol[b] = 0;
        for(i = 0; i < M; i++)
        {
            max_kol = 0;
            for(j = 0; j < N; j++)
            {
                if(k == 0)
                    suma[i] += zad[j][i];
                if(zad[j][i] > max)
                {
                    max = zad[j][i];
                    maxi = i;
                }
                if(zad[j][i] > max_kol)
                {
                    max_kol = zad[j][i];
                }
            }
            for(j = 0; j < N; j++)
            {
                if(zad[j][i] == max_kol)
                {
                    suma_kol[j] += max_kol;
                }
            }
        }
    }
}
```

```

    }
    suma[maxi] -= max;
    int maks_suma = 0;
    for(j = 0; j < N; j++)
    {
        if((zad[j][maxi] == max) && (suma_kol[j] > maks_suma))
        {
            maxj = j;
        }
    }
    for(j = 0; j < N; j++) zad[j][maxi] = 0;

    for(i = 0; i < M; i++) zad[maxj][i] = 0;
}

int s = 0;
for(i = 0; i < M; i++) s += suma[i];
printf("%d", s);
return 0;
}

```

2.

```

#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
const int MAXN = 36;
int main()
{
    int n;
    long long m;
    long long poeni[MAXN + 1];
    vector < long long > frakcija;

    cin >> n >> m;
    for (int i = 0; i < n; i++)
        cin >> poeni[i];

    int srednji = n / 2;
    for (int mask = 0; mask < (1 << srednji); mask++)
    {
        long long pomZbir = 0;
        for (int i = 0; i < srednji; i++)
            if (mask & (1 << i)) pomZbir += poeni[i];

        frakcija.push_back(pomZbir);
    }

    int koliko = frakcija.size();
    sort(frakcija.begin(), frakcija.end());

    long long rez = 0;

```

```

int ostalo = n - srednji;
for (int mask = 0; mask < (1 << ostalo); mask++)
    {
        long long pomZbir = 0;
        for (int i = 0; i < ostalo; i++)
            if (mask & (1 << i)) pomZbir += poeni[srednji + i];

        int levo = -1, desno = koliko, mid;
        while (levo + 1 < desno)
            {
                mid = levo + desno; mid = mid >> 1;
                if (pomZbir + frakcija[mid] >= m)
                    desno = mid;
                else
                    levo = mid;
            }

        rez += koliko - desno;
    }

cout << rez << endl;

return 0;
}

```

3.

```

#include <iostream>
#include <cstdio>
#define MAXN 1024

using namespace std;

int n;
int odg[MAXN << 1];
char p1[MAXN], p2[MAXN];
char stanje[MAXN][MAXN][3]; // 0 (brojac_za_0 = brojac_za_1),
// 1 (brojac_za_0 - brojac_za_1 = -1),
// 2 (brojac_za_0 - brojac_za_1 = 1)

```

```

int kodiraj(int k) {
    if(k == 0) return 0;
    if(k == -1) return 1;
    return 2;
}

```

```

int dekodiraj(int k) {
    if(k == 0) return 0;
    if(k == 1) return -1;
    return 1;
}

```

```

bool provera(int sled) {
    return (-1 <= sled && sled <= 1);
}

```

```

}

void pisi() {
    int i = n, j = n, k = 0;
    int razl = 0, sz = 0;
    while(!(i == 0 && j == 0 && k == 0)) {
        odg[sz] = stanje[i][j][k];

        if(odg[sz] == 1) razl -= p1[i - 1], i--;
        else razl -= p2[j - 1], j--;
        k = kodiraj(razl);

        sz++;
    }

    for(i=sz-1; i>=0; i--) printf("%d", odg[i]);
    printf("\n");
}

int main() {
    scanf("%d", &n);
    scanf("%s%s", &p1, &p2);

    int i, j, k;
    int razl, sled;

    for(i=0; i<n; i++)
    {
        if(p1[i] == '0') p1[i] = 1;
        else p1[i] = -1;

        if(p2[i] == '0') p2[i] = 1;
        else p2[i] = -1;
    }

    stanje[0][0][0] = 0;

    for(i=0; i<=n; i++)
    for(j=0; j<=n; j++)
    for(k=0; k<3; k++)
    if(stanje[i][j][k] || (i == 0 && j == 0 && k == 0)) {
        razl = dekodiraj(k);

        sled = razl + p1[i];
        if(i < n && provera(sled)) stanje[i + 1][j][kodiraj(sled)] = 1;

        sled = razl + p2[j];
        if(j < n && provera(sled)) stanje[i][j + 1][kodiraj(sled)] = 2;
    }

    pisi();
    return 0;
}

```

```

4.
#include <iostream>
using namespace std;
int s,n,i,p,w,k=1000000000,j;
int B[10000] ;

void minibure(int p,int T, int m)
{ //p je tekuci broj bureta, T je kapacitet nasutog vina u burice
  //m je broj upotrebljenih burica
  int br;
  br=(s-T)/B[p]; //max broj burica kapaciteta B[p]
  if (m+br<k) {
    if (T+br*B[p]==s)k=m+br; //proces se zaustavlja, k je min broj burica
      else if (p<n) //ako nismo upotrebili jos sve burice
        while (br>=0)
        {
          minibure(p+1, T+br*B[p],m+br);
          br--;
        }
    }
  }

int main()
{
  cin >> s>>n;
  for(i=1;i<=n;i++) cin >>B[i]; // burici B[1], B[2],..., B[n]
  for (i=n; i>=2; i--){
    p=1;
    for (j=2; j<=i; j++)
      if (B[p]>B[j])p=j;
    w=B[p]; B[p]=B[i]; B[i]=w;
  }
  minibure(1,0,0);
  cout <<k<<endl;
}

```