ELEKTRIJADA 2012

Kranevo (Bulgaria), May 18-23, 2012

INFORMATICS

Tasks

❶ **Find the output of the following program:**

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

char *fs(char *st, char *sp)
{
   int i, j;
   char *res=NULL, *t=NULL;
   if (!sp[0]) return (st);
   t = (char*) malloc(strlen(sp)+1);
   t[0] = -1;
   for (i=0; sp[i]; i++)
   {
      t[i+1] = t[i] + 1;
      while (t[i+1]>0 && sp[i]!=sp[t[i+1]-1])
         t[i+1] = t[t[i+1]-1] + 1;
   }
   for (i=j=0; st[i]; )
   {
      if (j<0 || st[i]==sp[j])
      {
         ++i, ++j;
         if (!sp[j])
         { res = st+i-j; break; }
      }
      else j=t[j];
   }
   free(t);
   return (res);
}

int main()
{
   int i;
   static char *s="ELEKTRIJADA", *res;
   for (i=0; s[i]; i++)
      res=fs(s,s+i);
   printf("%s", res);
   return (0);
}
```

❷ **Find the output of the following program:**

```
#include <stdio.h>
int x,c1,c2,sc2;

int f(int i)
{
   if (i && !((i-1)&i)) f((c1++,i>>1));
   return (c1);
}

int main()
{
   int i;
   for (i=1; i<10; i++, c1=0)
   {
      c2 += (x=f(i)) ? --x : x;
      if (sc2!=c2) printf("%d\n", sc2=c2);
   }
   return (0);
}
```

❸ **Find the output of the following program:**

```
#include <stdio.h>

#define X 15
#define Y X/2

int main()
{
   int xx, yy, p, q, tmp, min;
   static int b[50][50];
   for (p=1; p<=X; p++) b[p][1]=p;
     for (q=1; q<=Y; q++) b[1][q]=q;
       for (xx=2; xx<=X; xx++)
       {
         for (yy=2; yy<=Y; yy++)
         {
            if (xx==yy)
              b[xx][yy]=1;
            else
            {
               min=100;
               for (p=1; p<xx; p++)
               {
                 tmp = b[p][yy] + b[xx-p][yy];
                 if (tmp<min)
                    min = tmp;
               }
               for (q=1; q<yy; q++)
               {
                 tmp = b[xx][q] + b[xx][yy-q];
                 if (tmp<min)
                    min = tmp;
               }
               b[xx][yy] = min;
            }
         }
       }
   printf("%d", b[X][Y]);
   return (0);
}
```

❹ **Find the output of the following program:**

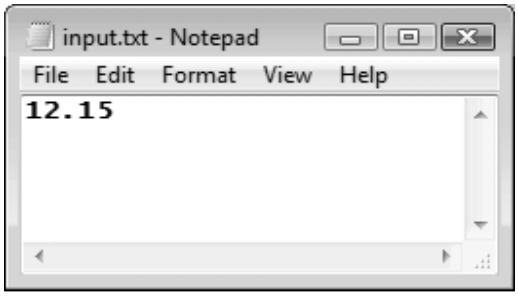
```
#include <stdio.h>
int x,c1,c2;

int f(int i)
{
   if (i && (i-1)&i == 0) f((c1++,i>>1));
   return (c1);
}

int main()
{
   int i;
   for (i=1; i<10; i++, c1=0)
      c2 += (x=f(i)) ? --x : x;
   printf("%d", c2);
   return (0);
}
```

❺ Find the output of the following program:

```
#include <stdio.h>

int main()
{
    int i, c, k, n;
    char *s="39";
    for (i=c=0, k=1; s[i]; k=1, c=0, i++)
    {
        int x[100] = { 0, 1 };
        while (k)
        {
           c++;
           if (x[k]==s[i]-'0')
              { k--; x[k]++; }
           else
              { k++; x[k]=x[k-1]+1; }
        }
        printf("%d\n", c);
    }
    return (0);
}
```

❻ Find the output of the following program (file **input.txt** is depicted bellow the code**):**

```
#include <stdio.h>
long n, m;

void fv(long cd, long bb, long nn)
{
   long ai = cd + bb / nn;
   printf("%d ", ai);
   bb %= nn;
   if (ai>m && n>0) m=ai;
   if (bb>0) { n++; fv(nn/bb, nn%bb, bb); }
}

int main()
{
  FILE *fin;
  if ((fin=fopen("input.txt","r"))!=NULL)
  {
    char c[100];
    long cd=0, bb=0, nn=1, i=0, rcd=1;
    fscanf(fin, "%s", c);
    for ( ; c[i]; i++)
      if (c[i]=='.') rcd=0;
      else
        if (rcd) cd=cd*10+c[i]-'0';
        else { bb=bb*10+c[i]-'0'; nn*=10; }
    fv(cd, bb, nn);
    printf("\n%d", m);
  }
  return (0);
}
```



❼ Find the output of the following program:

```
#include <stdio.h>

#define X(a,b) printf(a,b)

int main()
{
   static char s[]="INFORMATICS";
   1 || printf("A");
   0 || printf("B");
   1 && printf("C");
   0 && printf("D");
   X("%d",X("%o\n",X("\n%s\n",s)));
   return (0);
}
```

❽ Find the output of the following program:

```
#include <stdio.h>

typedef struct time_1
{
   unsigned int hh:4, mm:6, am:1;
} T1;

typedef struct time_2
{
   unsigned int hh:4, mm:6, :5, am:1;
} T2;

typedef struct time_3
{
   unsigned int hh:4;
   unsigned short int :0;
   unsigned int mm:6, am:1;
} T3;

int main()
{
   union u_1 { T1 t; int d; } u1; u1.d=0;
   union u_2 { T2 t; int d; } u2; u2.d=0;
   union u_3 { T3 t; int d; } u3; u3.d=0;
   u1.t.hh=11; u1.t.mm=31; u1.t.am=1;
   u2.t.hh=11; u2.t.mm=31; u2.t.am=1;
   u3.t.hh=11; u3.t.mm=31; u3.t.am=1;
   printf("%x\n", u1.d);
   printf("%x\n", u2.d);
   printf("%x", u3.d);
   return (0);
}
```

❾ **Find the output of the following program (**suppose that **sizeof(char *)** returns 4**):**

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

void **md2(int nu, int nd, int *pd);
void md3(char ***tip,int nu,int nd,int *pd);

int gu,te;

void *mdalloc(int nd, int w, int *pd)
{
   int i;
   char ***tip;
   if (pd == NULL) return (NULL);
   gu = w;
   tip = (char ***)md2(pd[0], nd, &pd[1]);
   if (nd>1 && tip)
      md3(tip, pd[0], nd-1, &pd[1]);
   return (tip);
}

void mdfree(void *tip, int nd)
{
   if (nd!=1)
      mdfree(((void **)tip)[0], nd-1);
   free(tip);
}

void **md2(int nu, int nd, int *pd)
{
  char **tip;
  if (nd==1)
    tip=(char**) malloc(nu*gu);
  else
  {
    tip=(char**) malloc(nu*sizeof(char *));
    if (tip)
    {
     tip[0]=(char *)md2(nu*pd[0],nd-1,&pd[1]);
     if (tip[0]==NULL)
     {
        free(tip); tip = NULL;
     }
    }
  }
  return ((void **)tip);
}

void md3(char ***tip, int nu, int nd, int *pd)
{
 int i;
 for (i=1; i<nu; i++,te++)
 {
  if (nd==1)
     tip[i]=
       (char **)((char *)tip[0]+i*pd[0]*gu);
  else
     tip[i]=tip[0]+i*pd[0];
 }
 if (nd>1)
   md3((char ***)tip[0],nu*pd[0],nd-1,&pd[1]);
}

int main()
{
   int a[]={2,3,4};
   int ***ma;
   ma = (int***) mdalloc(3,4,a);
   printf("%d\n%d", sizeof(ma), te);
   mdfree(ma,3);
   return (0);
}
```

❿ **Find the output of the following program:**

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

typedef struct ln{char c; struct ln *nx;} LN;
LN *lh, *lv;

void la(LN **q, int d)
{
  LN *t=*q; int x=0;
  if (*q==NULL)
     t=*q=(LN *)malloc(sizeof(LN));
  else
  {
    for ( ; t->nx ; t=t->nx)
      if (x=t->c==d) break;
    if (!x) t=t->nx=(LN *)malloc(sizeof(LN));
  }
  t->c=d; t->nx=NULL;
}

void lp()
{
   for (lv=lh; lv; lv=lv->nx) putchar(lv->c);
}

void lss()
{
   LN *a=lh,*b=NULL,*c=lh,*d=NULL,*t=NULL;
   while(a->nx)
   {
     d=b=a->nx;
     while (b)
     {
       if (a->c>b->c)
       {
         if (a->nx==b)
         {
           if (a==lh)
           {
             a->nx=b->nx; b->nx=a; t=a; a=b;
             b=t; lh=c=a; d=b; b=b->nx;
           } else {
             a->nx=b->nx; b->nx=a; c->nx=b;
             t=a; a=b; b=t; d=b; b=b->nx;
           }
         } else {
           if (a==lh)
           {
             t=b->nx; b->nx=a->nx; a->nx=t;
             d->nx=a; t=a; a=b; b=t; d=b;
             b=b->nx; lh=a;
           } else {
             t=b->nx; b->nx=a->nx; a->nx=t;
             c->nx=b; d->nx=a;
             t=a; a=b; b=t; d=b; b=b->nx;
           }
         }
       } else { d=b; b=b->nx; }
     }
     c=a; a=a->nx;
   }
}

int main()
{
   LN *nn=NULL; int i;
   char txt[]="INFORMATICS";
   for (i=0; txt[i]; i++) la(&nn, i[txt]);
   lh=nn; lp(); putchar('\n'); lss(); lp();
   return (0);
}
```