ELEKTRIJADA 2014

Siofok (Hungary), May 7-12, 2014

INFORMATICS

Tasks

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

```
#include <stdio.h>
enum Rim { I, II = I + I, III, V = III + II,
IV = 4, VI, VII = VI + I };
int main(){
  char c=074, d=110;
  do{
    d += VII; c += V;
    printf("%3d %3d\n", c, d);
  } while (c + d > (char) 0);
  return 0;
}
```

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

```
#include <stdio.h>
int main()
{
  int a, b, c = 4;
  char text[] = "ELEKTRIJADA 2014 SIOFOK \
HUNGARY 1420 2013 1320 HUNGARY HUNGARY \
0705 1205 BALATON";
  char *t = text, s[50] = { 0 };
  while (sscanf(t, "%[^HUNGARY]%*s%2d%2d",
         s, &a, &b) == 3 && *s){
    if (--c <= 0)
      c = a - b;
    printf("%4s\n", s + printf("%0*d", c, b));
    t += 7;
  }
  return 0;
}
```

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

```
#include <stdio.h>
#include <stdio.h>
int *fb(int p[][5], int *q)
{
  int i, y, x = *--q / **p++;
  for (i = 0; i < 3; i++, (**p)++);
  y = p[-1][-4];
  printf("%d %d", x, y);
  return ((int*)p);
}

int main()
{
  int i, j;
  int a[][5]={{11,10,9},{8,7,6},{5,4,3},{2}};
  int b[] = {10,20,30,40,50,60,70,80,90};
  printf(" %d\n", 2[fb(a+1, (int*)(&b + 1))]);
  for (i = 0; i < 3; i++)
    for (j = i + 2; j < 4; j++)
      printf("%d ", a[i, j] - a[j, i]);
  printf("\n");
  for (i = 0; i < 3; i++)
    for (j = i + 1; j < 3; j++)
      printf("%d %d\n", i[j[a]], j[i[a]]);
  return 0;
}
```

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

```
#include <stdio.h>
#define max(a,b) ((a) > (b) ? (a) : (b))
int e;
int f(int n)
{
  static int x = 1;
  int e = 1;
  if (x > max(--n, e)){
    extern int e;
    ++e; x += max(--n, e);
  }
  if (n > 0)
    x += f(--n), f(--n);
  return x;
}
int main()
{
  int i;
  for (i = 1; i <= 4; i++)
    printf("%d ", f(i));
  return 0;
}
```

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

```
#include <stdio.h>
int x;
void f(char ***p1, char ***p2, char **p3)
{
  if (***p1****p2***p3){
    ++x;
    ++*p1;
    ++**p2;
    ++**p3;
    f(p1, p2, p3);
  }
}
int main()
{
  char *s1[] = { "2014", "welcome", "to",
                 "siofok", "hungary", "" };
  char *s2[] = { "informatics", "good",
           "luck", "in", "competition", "" };
  char s3[][8] = { "have", "a nice", "time",
                 "in", "the", "evening", "" };
  char **p1 = s1, **p2 = s2, *p3 = s3[0];

  printf("%d %d %d\n",
      sizeof(s1), sizeof(*s1), sizeof(**s1));

  printf("%d %d %d\n",
      sizeof(s3), sizeof(*s3), sizeof(**s3));

  f(&p1, &p2, &p3);

  printf("%d %s %s %s\n", x, *s1, *s2, *s3);
  return (0);
}
```

❻ Find the output of the following program:

```
#include <stdio.h>
#define M 6
#define MAX(p) ((p.r) > (p.s) ?\
(p.rs = 1, p.r) : (p.rs = 0, p.s))
#define M1(a) ((a) ? --a : a)
typedef struct{ int r, s; char rs; } pair;
int* rs(int *m, int i, int j, int s){
  return m + i * s + j; }
int* sr(int *m, int i, int j, int s){
  return m + j * s + i; }
void m(int c[][M],int*(*f)(int*,int,int,int)){
  int i, j, p, v;
  for (i = 0, p = 0; i < M; i++){
    for (j = 1; j < M; j++)
      if (*f(c[0],i,j,M) < *f(c[0],i,p,M))
        p = j;
    v = *f(c[0], i, p, M);
    for (j = 0; j < M; j++)
      *f(c[0], i, j, M) -= v;
  }
}
int cr(int c[][M], pair *z, pair *cd){
  int i,p, max, cnt = 0;
  for (i = 0; i < M; i++) cd[i].r = cd[i].s=0;
  do{
    max = MAX(z[p=0]);
    for (i = 1; i<M; i++)
      if (MAX(z[i]) > max){
        max = MAX(z[i]); p = i;
      }
    if (max){
      cnt++; printf("%c%d-%d%c",
       z[p].rs ?'R':'S',p,max,cnt%2?' ':'\n');
      for (i = 0; i < M; i++)
        if (! c[z[p].rs?p:i][z[p].rs?i:p] ){
          z[p].rs ? M1(z[i].s) : M1(z[i].r);
        }
      if (z[p].rs) {
        z[p].r = 0; cd[p].r |= 1;
      } else {
        z[p].s = 0; cd[p].s |= 1;
      }
    }
  } while (max);
  return cnt;
}
int main() {
 int i,j,t,c[M][M] = {{ 55,99,20,96,73,74},
 { 51,36,68,91,30,36},{ 74, 2, 8,53,26, 9},
 { 31,73, 1,27,90, 1},{ 46,72,57,70,28,77},
 {  7, 8, 2,57,81,52}};
  pair pairs[M] = { {0} }, cd[M];
  m(c, rs);  m(c, sr);
  for (i = 0; i < M; i++)
    for (j = 0; j < M; j++)
      if (!c[i][j]){
        pairs[i].r++; pairs[j].s++;
      }
  if(!cr(c, pairs, cd) != M){
    for (t = i = 0; i < M; i++){
      if (cd[i].r) continue;
      for (j = 0; j < M; j++){
        if (cd[j].s) continue;
        if (!t || c[i][j] < t) t = c[i][j];
      }
    }
    printf("\n%d ", t);
  }
  return 0;
}
```

❼ Find the output of the following program:

```
#include <stdio.h>
#include <stdlib.h>
struct _node{ int a,b; struct _node *l, *r; };
typedef struct _node node;

int sumA(node *r){
  return r ? r->a-'D'+sumA(r->l)+sumA(r->r):0;
}
int sumB(node *r){
  return r ? r->b + sumB(r->l)+sumB(r->r) : 0;
}
int cnt(node *r){
  return r ? 1 + cnt(r->l) + cnt(r->r) : 0;
}

void setB(node *root){
  node *p[5];
  p[0] = root;
  while (p[0]){
    p[1] = p[0];
    p[2] = root;
    do{
      for (p[3]=p[0],p[4]=p[2]; p[3] && p[4];
           p[3] = p[3]->r, p[4] = p[4]->l){
        p[1]->b += (p[3]->a - 'D')
                 * (p[4]->a - 'D');
      }
      p[1] = p[1]->r; p[2] = p[2]->r;
    } while (p[1]);
    p[0] = p[0]->l;
  }
}

int main(){
  node *n, *root=NULL, *p=NULL, *ln=NULL;
  int i;
  char split = 'B', h=0;
  char data[] =
{'E','B','C','C','D','D','E','E','D','B',
'D','F','F','E','C','D','D','D','F','C' };

  for (i = 0; i<sizeof(data); h = data[i++]){
    if (data[i] == h + 1) continue;
    n = (node*)malloc(sizeof(node));
    n->a = data[i]; n->b = 0;
    n->l = n->r = NULL;
    if (data[i] == 'D' + split){
      split = (split + 1) % 3;
      for (ln = root; ln; p = ln, ln = ln->l);
    }
    if (p != NULL){
      p->l = n; p = p->r;
    }
    if (root == NULL){
      root = n; split = root->a - 'D';
    }
    if (ln == NULL) ln = n;
    else{
      ln ->r = n; ln = ln ->r;
    }
  }

  setB(root);
  printf("%d\n%d\n%d",
       cnt(root), sumA(root), sumB(root));
  return 0;
}
```

❽ Find the output of the following program:

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
#define isnull(x) ((x) == NULL ? 0 : 1)
#define bits(a, b, c) \
 (isnull(a)<<2) + (isnull(b) << 1) + isnull(c)
typedef struct _n{ char v; struct _n *l, *r;
} node;

node* pp(node *p, node *q){
  if (p->v < q->v){
    p->r = q->l; q->l = p; return q;
  }
  else{
    q->r = p->l; p->l = q; return p;
  }
}

void ms(node **a, node **b, int max){
  int i;
  node *c = NULL;
  for (i = 0; i < MAX && i<max; i++){
    switch (bits(c, b[i], a[i])){
      case 2: a[i] = b[i]; break;
      case 3: c = pp(a[i], b[i]);
              a[i] = NULL; break;
      case 4: a[i] = c; c = NULL; break;
      case 5: c = pp(c, a[i]);
              a[i] = NULL; break;
      case 6: case 7: pp(c, b[i]); break;
    }
  }
}

node *uu(node **nodes, char val){
   int i; node *c, *t;
   c = t = (node*)malloc(sizeof(node));
   c->l = c->r = NULL;
   c->v = val;
   for (i = 0; i < MAX; i++){
     if (c == NULL) break;
     if (nodes[i] == NULL){
       nodes[i] = c;
       break;
     }
     else{
       c = pp(c, nodes[i]);
       nodes[i] = NULL;
     }
   }
   return t;
}

void dump(node *root){
  if (root) {
    printf("%c", root->v);
    dump(root->l);     dump(root->r);
  }
}
void dump_p(node **head, int cnt){
  int i, prev = 0;
  for (i = 0; i < cnt; i++){
    if (head[i]){
      printf("%s%d.", prev++ ? " " : "",  i);
      dump(head[i]);
    }
  }
  printf("\n");
}
```

```
int main(){
  char *s = "Informatics";
  node *p1[MAX] = { 0 }, *p2[MAX] = { 0 };
  while (*s) uu(p1, *s++);
  dump_p(p1, 10);
  while(*--s != 'I') uu(p2, *s - 32);
  ms(p1, p2, 10);
  dump_p(p1, 5);
  return 0;
}
```

❾ Find the output of the following program:

```
#include <stdio.h>

int f(int n){
  int r = 1;
  if (n > 1){
    int x = f(n - 1);
    int y = f(n - 2 & n - 1);
    r = (x&y) + ((x^y) >> 1);
    r += ((x^y) + 2 * (x&y)) - 1;
  }
  return r;
}

int main(){
  printf("%d", f((8,7,6)));
  return 0;
}
```

❿ Find the output of the following program:

```
#include <stdio.h>
#define F(a,b,c,d,e,f) a##f##d(#b,#e,0##c##12)

int g, c12 = 12;
char *s = "ELEKTRIJADA 2014";

int t(int i, int j){
  unsigned char p1 = 0125, p2 = 025;
  p1 &= ~(0xF * 0x10);
  p2 |= 0xF * p1 / (i-j+1);
  g |= ~(~p1^~p2);
  return 'T' - s[p2+g];
}

void st(int i, int j){
  unsigned int u = ~1;
  g ^= (u | i) & j;
  g = -g / (1 << 3);
}

int test(char *a, char *b, int i){
  printf("%s %s %d\n", a, b, i);
  return *b - *a;
}

int main(){
  int a, e = 5, x = 10, oldg;
  a = F(t, 13, x, st, 9, e);
  oldg = g;
  g ^= 0x000A;
  printf("%d", test(s + a, s + g, oldg));
  return 0;
}
```

Appendix: ASCII table

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del