

ELEKTRIJADA 2015

Bečići (Montenegro), May 19-24, 2015

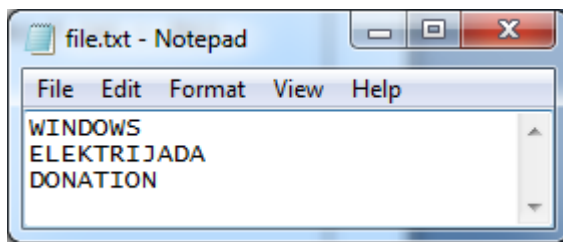
INFORMATICS

Tasks

1 Find the output of the following program:

```
#include <stdio.h>
#define MAX 20
int main()
{
    signed char niz[MAX][MAX], c=0;
    int d=1, x=0, i, j;
    FILE *f=fopen("file.txt", "r");
    while (fscanf(f, "%s", niz[x++])!=EOF);
    for (c=--x; c++&*niz; d<=1, d+=!(c<x>0));
    d -= c;
    for (i=0; i<x;
        i==1?printf("%c\n", niz[i][x>>1]):i, ++i)
        for (j=0; j<=i; printf("%c", niz[i][j++]));
    printf("%X", d);
    return 0;
}
```

Assume the following input file:



2 Find the output of the following program:

```
#include <stdio.h>
#define RS(a,b) a+=b, b=a-b, a-=b
int f(int x) { return x<10?x:x%10+f(x/10); }
void fs(int a[], int ib, int ie)
{
    if (ib<ie)
    {
        int i=ib, j=ie, pi=a[ib];
        while (i<j)
        {
            while (f(a[i])>=f(pi) && i<j) i++;
            while (f(a[j])<f(pi)) j--;
            if (i<j) RS(a[i],a[j]);
        }
        a[ib]=a[j]; a[j]=pi;
        fs(a,ib,j-1); fs(a,j+1,ie);
    }
}
int main()
{
    int i=0, niz[10]={2015,19,5,2016,20,6};
    fs(niz,0,5);
    while (niz[i]) printf("%d\n", niz[i]), i++;
    return 0;
}
```

3 Find the output of the following program:

```
#include <stdio.h>
#include <stdlib.h>
int ***fc(int n)
{
    int i,j,k;
    int ***p=(int***) malloc(n*sizeof(int **));
    for (i=0; i<n; i++)
    {
        *(p+i)=(int**) malloc(n*sizeof(int *));
        for (j=0; j<n; j++)
        {
            (*(p+i)+j)=
                (int*) malloc(n*sizeof(int));
            for (k=0; k<n; k++)
                *((*(p+i)+j)+k)=n*(n*i+j)+k;
        }
    }
    return p;
}
int ***ft(int ***q, int n)
{
    int i,j,k,x;
    for (x=***q, i=0; i<n; i++)
        for (j=0; j<n; j++)
            for (k=0; k<n; k++)
                if (i[j[k[q]]]>x && i[j[k[q]]]%2)
                    printf("%o\n", x=i[j[k[q]]]);
    return q;
}
void ff(int ***r, int n)
{
    int i,j;
    for(i=0; i<n; free(*(r+i)), i++)
        for(j=0; j<n; free(*(r+i)+j), j++)
            free(r);
}
int main()
{
    ff(ft(fc(3),3),3);
    return 0;
}
```

4 Find the output of the following program:

```
#include <stdio.h>
int main()
{
    char *s[]={"PERFECT", "MISC", "TUBE", "GREEN"};
    int i,j;
    for(i=0; i<sizeof(s)/sizeof(char*); i++)
        for(j=0; j<sizeof(s[i])/sizeof(char); j++)
            j+i&0123>>i||printf("%c", j[i[s]]);
    return 0;
}
```

5 Find the output of the following program:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#define RE(a,b) \
    realloc(a,strlen(a)+strlen(b)+1)
#define TO(z,a,b,c) z(a==b)?z(a==b):z(a==c)
#define VA(a,b) va_arg(a,b)
char *sf(char *ks, ...)
{
    int f(int x) { return x?f(x/0x0a)+1:x+1; }
    char *v, *p, *ts, *r=(char *)calloc(1,1);
    int d, i, li;
    va_list val;
    va_start(val, ks);
    for (p=ks; *p; ++p, i=0)
        if (d = TO(+,*p,'#',TO(-,*p,'%','%'))
            {
                v=(char *)calloc(li=f(VA(val,int)),1);
                ks = p-ks < li-1 ? ks : p-li+1;
                for (d--,ts=p-1; ks!=ts; d?ks++:ts--)
                    v[i++] = d ? *ks : *ts;
                v[i]=*ks;
                r = (char *)RE(r,v);
                strcat(r,v);
                ks=p+1;
            }
    va_end(val);
    return r;
}
int main()
{
    int i;
    char *s[] = {"ELEKT%RI#JADA%",
                "MONTE#NEG#RO%"};
    for(i=0; i<2; i++)
        printf("%s\n", sf(s[i],2015,2016,1));
    return 0;
}
```

6 Find the output of the following program:

```
#include <stdio.h>
void main()
{
    signed char x[]={-256,-255,-129,-128,-127,
                    -1,127,128,255,256};
    signed char fv[10]={}, fc[10]={};
    int i,j,m=-1;
    for (i=0;i<sizeof(x); i++)
        if (x[i]>100u)
            {
                for (j=0;j<m;j++)
                    if (x[i]==fv[j]) break;
                fv[j]=x[i],fc[j]++;
                if (j>=m) m++;
            }
    for(j=0; j<m; j++)
        printf("%d(%d)\n", fv[j],fc[j]);
}
```

7 Suppose that "program.exe" is the corresponding executable of the following source code:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define F(e) "ERR: "#e
#define T(e) "REZ: "#e
int fs(char *s, char *p)
{
    static int br;
    if (s=strstr(s,p)) br++, fs(s+strlen(p),p);
    return br;
}
int main(int argc, char* argv[])
{
    char *p=0, *s=0, *psn=0;
    int i, n=1;
    for (i=1; i<argc; i++)
    {
        if (!strcmp(argv[i],"-p"))
        {
            if (++i<argc) p=argv[i];
            else return printf(F(p)), 1;
        }
        else if (!strcmp(argv[i],"-s"))
        {
            if (++i<argc) s=argv[i];
            else return printf(F(s)), 1;
        }
        else if (!strcmp(argv[i],"-n"))
        {
            if (++i<argc) n=atoi(argv[i]);
            else return printf(F(n)), 1;
        }
        else return printf(F(psn)), 1;
    }
    if (!p) return printf(F(p)), 1;
    if (!s) return printf(F(s)), 1;
    if (fs(s,p)!=n)
        return printf(F(n)"=%d",fs(s,p)), 1;
    else
        return printf(T(n)"=%d",fs(s,p)), 0;
}
```

Find the output, if the command line looks like:

d:\>program -s "ELEKTRIJADA BECICI" -p E -n 2

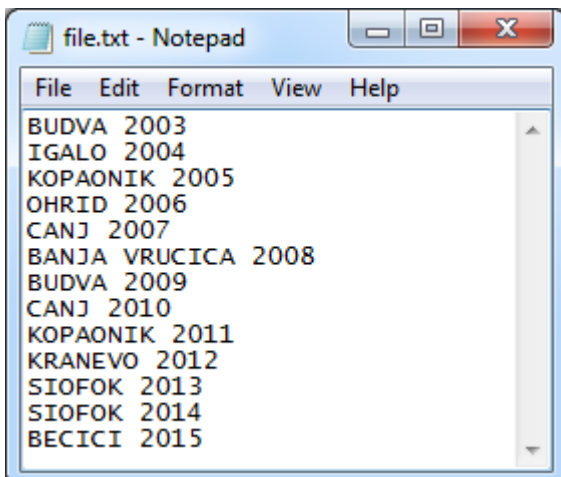
8 Find the output of the following program:
(suppose that `sizeof(char *)` returns 4):

```
#include <stdio.h>
typedef struct S
{
    void (*foo[24])(void*);
    char *****(*bar)(void*);
} SA[10];
void main()
{
    SA array[10];
    printf("%d\n%d", sizeof(SA),sizeof(array));
}
```

9 Find the output of the following program:

```
#include <stdio.h>
#include <stdlib.h>
#define QN (NODE*)malloc(sizeof(NODE))
typedef struct node
{ char c; int i; struct node *n,*u; } NODE;
void add(NODE **p, char *s)
{
    NODE *q;
    if (*s)
        if (*p==NULL)
            *p=QN, (*p)->c=*s, (*p)->i=1, (*p)->u=NULL,
            (*p)->n=QN, (*p)->n->n=*p, q=*p=(*p)->n,
            (*p)->c=*s, (*p)->i=1, (*p)->u=NULL;
        else
        {
            q=(*p)->n->n;
            while (q->c!=*s && q!=*p) q=q->n;
            if (q->c!=*s)
                q=QN, q->c=*s, q->i=1, q->u=NULL,
                q->n=(*p)->n, *p=(*p)->n=q, q->n->i++;
            else q->i++;
        }
    for(s++;*s;s++)
    {
        if (q->u) q=q->u;
        else q=q->u=QN, q->u=NULL;
        q->c=*s; q->i=1;
        if (q->u) q->u->i=0;
    }
}
int main()
{
    NODE *f=NULL, *q;
    char w[100];
    int i=2;
    FILE *fp=fopen("file.txt", "r");
    while (fscanf(fp,"%s",w)!=EOF) add(&f,w);
    while (f->u) f=f->u;
    printf("%d", f->i);
    while(i--)
    {
        q=f->n; printf("\n%d ", q->i);
        while (q)
        {
            if (!q->i) break;
            printf("%c", q->c); q=q->u;
        }
    }
    return 0;
}
```

Assume the following input file:

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

```
#include <stdio.h>
#include <stdlib.h>
struct N { char c; int n; struct N **p; };
typedef struct N NODE;

NODE *bt, *f;
NODE *newNode(char c)
{
    NODE *q = (NODE*) malloc(sizeof(NODE));
    q->p=NULL; q->n=1; q->c=c;
    return q;
}
NODE *fa(NODE *q, char s)
{
    if (!q) q=newNode(s);
    else if (q->c==s) q->n++;
    else
    {
        if (!q->p)
            q->p=(NODE**) calloc(2, sizeof(NODE*));
        if (s<q->c) *(q->p)=fa(*(q->p), s);
        else *(q->p+1)=fa(*(q->p+1), s);
    }
    return q;
}
void ft(NODE *q)
{
    if (q)
    {
        NODE **p;
        if (q->p) ft(*(q->p));
        printf("%c:%d\n", q->c, q->n);
        if (q->p) ft(*(q->p+1));
    }
}
NODE *fc(NODE *q, char *s)
{
    if (!q) q=newNode(*s++);
    for(;*s;s++)
        if (!q->p)
        {
            q->p=(NODE**) calloc(2, sizeof(NODE*));
            *(q->p)=newNode(*s);
            if (*(s+1)) fc(*(q->p), s+1);
        }
    else
    {
        int i,n;
        NODE **tmp=q->p;
        for (n=1; *tmp; tmp++, n++);
        tmp = (NODE**) calloc(n+1, sizeof(NODE*));
        for (i=0; i<n; i++)
            *(tmp+i)=*( (q->p)+i );
        *(tmp+i-1)=newNode(*s);
        if (*(s+1)) fc(*(q->p), s+1);
        free(q->p);
        q->p=tmp;
    }
    return q;
}
void fx(NODE *q)
{
    NODE **p;
    if (q)
    {
        bt=fa(bt, q->c);
        if (q->p) for(p=q->p; *p; p++) fx(*p);
    }
}
void main() { fx(fc(f, "BECICI")); ft(bt); }
```

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