

Писмен испит по

Структурирано програмирање

Скопје, 23.08.2010

1. Да се напише програма која од непознат број **парови броеви** ќе ги изброи и испечати оние парови за кои е исполнет следниот услов: бројот запишан со истите цифри, но во обратен редослед од првиот број да е **поголем** од бројот запишан со истите цифри, но во обратен редослед од вториот број. Претворањето на даден број во број запишан со истите цифри, но во обратен редослед да се реализира со посебна функција.

Пример: 1. влез: 27 и 13 -> 72 > 31 => условот е исполнет
2. влез 1753+3467 -> 3571 < 7643 => условот не е исполнет

25 поени

```
int prevrtiBr(int broj);
int main ()
{
    int a, b, br = 0;
    while (scanf("%d %d", &a, &b) == 2)
    {
        if (prevrtiBr(a) > prevrtiBr(b))
        {
            br++;
            printf("%d %d\n", a, b);
        }
    }
    printf("Vкупno %d broevi go ispolnuvaat uslovot.\n", br);

    return 0;
}

int prevrtiBr(int broj)
{
    int nov = 0;
    while (broj)
    {
        nov = nov*10+broj%10;
        broj /= 10;
    }
    return nov;
}
```

2. Да се напише функција која прави споредба на знаците на соодветни позиции од две текстуални низи кои функцијата ги прима на влез. Ако соодветните знаци се исти, тогаш тие се заменуваат со знакот * во двете низи, а во спротивно, остануваат непроменети. Како резултат од функцијата да се врати бројот на знаци во кои низите се совпаѓаат. Должината на двете низи може да биде различна. Задачата да се реши со помош на **покажувачи**.

Пример: прва низа: Jas sum prva niza
втора низа: Zadaca2 po programiranje

Низите се совпаѓаат во 4 знаци.

прва низа: J*s sum**rva niz*

втора низа: Z*daca2**o progr*miranje

25 поени

```
int sporedi(char*str1, char*str2)
{
    int br = 0;
    while ((*str1!='\0') && (*str2!='\0'))
    {
        if (*str1 == *str2)
        {
            br++;
            (*str1)++;
            (*str2)++;
        }
    }
    return br;
}
```

```

        *str1 = *str2 = '*';
        br++;
    }
    str1++; str2++;
}
return br;
}

```

3. Да се напише програма во која од тастатура се внесуваат елементите (целобројни вредности) на матрица со димензија $m \times n$ (не поголеми од 100). За секој елемент од матрицата се пресметува збирот на соседите по дијагонала. Ако елементот е поголем од пресметаниот збир, тогаш тој ја добива вредноста на збирот на неговите дијагонални соседи. Во спротивно, вредноста на елементот останува непроменета.

Пример:

	0	1	2	3
0	1	9	2	7
1	4	10	4	1
2	1	6	3	8
3	11	5	7	1

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

- На елементот 10 соседи по дијагонала се елементите 1, 2, 1 и 3;
- На елементот 9 соседи по дијагонала се елементите 4 и 4;
- На елементот 11 сосед по дијагонала е елементот 6.

25 поени

```

int main ()
{
    int a[MAX][MAX], b[MAX+2][MAX+2], i, j, m, n, zbir = 0;

    printf("Vnesete gi dimenziite na matricata\n");
    scanf("%d %d", &m, &n);
    printf("Vnesete ja matricata\n");
    for (i=0; i<m; i++)
        for (j=0; j<n; j++)
        {
            printf("a[%d,%d]=", i, j);
            scanf("%d", &a[i][j]);
        }
    for (i=0; i<m+2; i++)
        for (j=0; j<n+2; j++)
        {
            b[i][j] = 0;
        }
    for (i=1; i<m+1; i++)
        for (j=1; j<n+1; j++)
        {
            b[i][j] = a[i-1][j-1];
        }

    for (i=1; i<m+1; i++)
        for (j=1; j<n+1; j++)
        {
            zbir = b[i-1][j-1] + b[i-1][j+1] + b[i+1][j+1] + b[i+1][j-1];
            if (b[i][j] > zbir)
                b[i][j] = zbir;
        }
    for (i=1; i<m+1; i++)
    {
        printf("\n");
        for (j=1; j<n+1; j++)
        {
            printf("%d\t", b[i][j]);
        }
    }
}

```

```

    }
}

return 0;
}

```

4. Да се напише програма која секој ред од дадена текстуална датотека ќе го копира во излезна датотека така што пред секој ред ќе додаде уште еден ред во кој ќе стои бројот на зборови што ги содржи прочитаниот ред од влезната датотека. Името на влезната и излезната датотека се задаваат како аргументи на командна линија, а ако не се зададени да се испечати начин на употреба на програмата. Зборовите се составени од алфанумерички знаци и се одделени со произволен број празни места и специјални знаци.

25 поени

```

int main(int argc, char* argv[])
{
    char s[MAX], *c;
    int vozbor = 0, brZborovi = 0;
    FILE *vleзна, *izleзна;
    if (argc==2)
    {
        printf("Upotreba:           %s           ime_na_vleзна_datoteka\n", argv[0]);
        return(-1);
    }
    if ((vleзна=fopen(argv[1], "r"))==NULL)
    {
        printf("Ne mozam da ja otvoram datotekata %s\n", argv[1]);
        return(-1);
    }
    if ((izleзна=fopen(argv[1], "w"))==NULL)
    {
        printf("Ne mozam da ja otvoram datotekata %s\n", argv[2]);
        return(-1);
    }
    while ((fgets(s, MAX, vleзна)) != NULL)
    {
        c=s; vozbor=0; brZborovi = 0;
        while (*c!='\0')
        {
            if (isalnum(*c))
            {
                if (!vozbor) vozbor=1;
            }
            else
            if (vozbor)
            {
                vozbor=0;
                brZborovi++;
            }
            c++;
        }
        if (vozbor)
            brZborovi++;
        fprintf(izleзна, "%d\n", brZborovi);
        fputs(s, izleзна);
    }

    return 0;
}

```

Испитот трае 120 мин

