

Втор парцијален испит по
СТРУКТУРИРАНО ПРОГРАМИРАЊЕ
28.11.2008
Група 1

1. Да се напише функција `double f(double x)` која ја пресметува функцијата:

$$J(x) = \sum_{k=0}^{10} \frac{\left(-\frac{1}{4}x^2\right)^k}{k!}$$

Пресметувањето на степен на број и факториел да се реализира со посебни рекурзивни функции.

```
int faktoriel(int n)
{
    if (n==0)
        return 1;
    return faktoriel(n-1)*n;
}

double stepen (double osn, int exp)
{
    if(exp == 0)
        return 1;
    return osn * stepen(osn, exp-1);
}

double f(double x)
{
    int k;
    double suma = 0.0;
    for(k = 0; k <= 10; k++)
        suma += stepen(-(x*x)/4, k)/faktoriel(k);
    return suma;
}
```

2. Да се напише функција која од дадена низа знаци (`str1`) ќе се направи нова низа знаци (`str3`) така што во првата низа ќе вметне друга низа знаци (`str2`) почнувајќи од дадена позиција (`pos`) .

На пример:

`str1`: "jas sum zadaca"

`str2`: "super "

`pos`: 8

`str3`: "jas sum super zadaca"

```
void vmetni(char*str1, char*str2, char*str3, int pos)
{
    strncpy(str3, str1, pos);
```

```

    strncat(str3, str2, strlen(str2));
    strcat(str3, str1 + pos);
}

int main()
{
    char s[MAX], d[MAX];
    printf("Vnesete niza:\n");
    gets(s);
    vmetni(s, "super ", d, 8);
    puts(d);
    system("pause");
    return 0;
}

```

Подолу е решението ако ја решат задачата без да формираат нова низа знаци:

```

void vmetni(char*str1, char*str2, int pos)
{
    char*s = str1 + strlen(str1);
    char*d = str1 + strlen(str1) + strlen(str2);
    *(d+1)='\0';

    for(;s != str1 + pos - 1; d--,s--)
    {
        *d = *s;
    }
    s++;
    while(*str2!='\0')
    {
        *s = *str2;
        s++; str2++;
    }
}

```

3. Да се напише програма која ќе прочита матрица A со димензии $m \times n$ (максималната вредност на m и n може да биде 100). Потребно е да се креира нова матрица B која во првата колона ќе ги содржи максималните елементи од соодветните редици во матрицата A , а во втората колона – позицијата на колоната во која се наоѓа секој од максималните елементи, соодветно. На крај треба да се испечати новата матрица B .

Пример:

	0	1	2	3		0	1
0	3	-2	5	1	0	5	2
1	-7	1	0	6	1	6	3
2	1	7	2	-8	2	7	1
3	4	-3	1	2	3	4	0

```

int main()
{
    int m,n,i,j,k;
    int a[MAX][MAX], max[MAX][2], t;
    printf("Vnesi gi dimenziite na matricata\n");
    scanf("%d %d",&m, &n);
    printf("Vnesi 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; i++)
    {
        max[i][0]=a[i][0];
        max[i][1]=0;
        for (j=1; j<n; j++)
            if (max[i][0]<a[i][j])
            {
                max[i][0]=a[i][j];
                max[i][1]=j;
            }
    }
    for (i=0; i<m; i++)
    {
        printf("\n");
        for (j=0; j<2; j++)
            printf("%d\t",max[i][j]);
    }

    return 0;
}

```

4. Да се напише програма која ќе го испечати бројот на појавувања на даден збор составен само од букви во текстуална датотека. Името на датотеката и бараниот збор се задаваат како аргументи од командна линија. Програмата проверува дали на командна линија се зададени сите потребни аргументи, а ако не се, тогаш печати порака “Nevalidni argument na komandna linija”. Се смета дека зборовите се составени само од букви и меѓусебно се разделени со произволен број на белини, специјални знаци и цифри. Во текстуалната датотека нема зборови подолги од 20 знаци.

Бонус: Задачата да се реши без креирање на текстуална низа за зборовите во текстуалната датотека.

```

int main(int argc, char** argv)
{
    char c;
    int vozbor=0, i, ok, brPojavuvanja = 0;
    FILE *dat;

    printf("%c\n",argv[1][0]);
}

```

```

printf("%c\n",argv[1][1]);
printf("%c\n",argv[1][2]);

if (argc!=3) {
    printf("Nevalidni argumenti na komandna linija\n");
    return(-1);
}
else
{
    if((dat=fopen(argv[1],"r"))==NULL)
    {
        printf("Ne mozam da ja otvoram datotekata %s\n",argv[1]);
        return(-1);
    }
    while((c=fgetc(dat))!=EOF) /* citaj znak po znak se do krajot */
    {
        if (isalpha(c))
        {
            if (!vozbor)
            {
                vozbor=1;
                i = 0;
                ok = 1;
            }
            if(argv[2][i]!=c)
                ok = 0;
            i++;
        }
        else
            if (vozbor) /* ako do sega sme bile vo zbor */
            {
                vozbor=0;
                if (ok)
                    brPojavuvanja++;
            }
    }
    if (vozbor && (ok))
        brPojavuvanja++; /* za na kraj */
    printf("Zborot %s se pojavuva %d pati vo datotekata\n", argv[2], brPojavuvanja);
}

return 0;
}

```

време за работа ***120 минути***