

ПРВ ПАРЦИЈАЛЕН ИСПИТ ПО
ПРОГРАМИРАЊЕ И АЛГОРИТМИ

20.11.2021

Задача 1 – Група 1

Да се напише програма за закажување на третман во салон за убавина. Корисникот прво треба да внесе колку пари има на располагање, а потоа програмата треба, врз основа на табелата дадена подолу, да пресмета каков третман може да добие корисникот (според тоа колку пари има).

Програмата треба прво да провери дали има вкупно пари да го купи задолжителниот подтретман. Доколку има, тогаш се печати порака дека може да се закаже третман, инаку, се печати дека не може да се закаже третман. Доколку може да се закаже третман, проверува дали има пари за сауна и или третман на тело. Доколку има пари само за сауна, тогаш се печати дека третманот ќе содржи сауна, или доколку има пари за третман на тело, тогаш се печати дека третманот ќе содржи и третман на тело (ако има пари за било кое од двете незадолжителни ставки, но нема истовремено за двете тогаш да го одбере поскапиот подтретман).

Доколку има пари и за двата незадолжителни подтретмани, тогаш се печати дека третманот ќе биде комплетен.

подтретман	Цена	Задолжително
Масажа	1000	ДА
Сауна	500	НЕ
Третман на тело	1000	НЕ

Решение:

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

```
int main(){
```

```
    int pari;
```

```
    printf("Vnesi kolku pari imash:");
```

```
    scanf("%d",&pari);
```

```
    if(pari>=1000){
```

```
        printf("Moze da se napravi rezervacija");
```

```
        pari-=1000;
```

```
    if(pari<500){printf("Nema dovolno pari za nitu eden od nezadolzitelните tretmani");}
```

```

else if(pari<1000){
    printf("Moze da zakaze sauna\n");
}
else if(pari<1500){
    printf("Moze da zakaze samo tretman za telo");
}else {
    printf("kompleten tretman\n");
}
}else {
    printf("ne moze da se napravi rezervacija");
}
return 0;
}

```

Задача 1 – група 2

Да се напише програма за организација на посета на антички локалитети во Македонија со помош на туристички водич. Корисникот прво треба да внесе колку пари има на располагање, а потоа програмата треба, врз основа на табелата дадена подолу, да пресмета колку долга може да биде посетата (колку може да трае). Програмата треба прво да провери дали корисникот има доволно средства за да направи задолжителната тура.

Доколку истата може да се направи, тогаш се зема во предвид времето на таа тура, и се проверува дали може да се направат и останатите тури. Ако нема доволно средства за задолжителната тура, не се прави ниедна тура и должината на посетата се поставува на вредност 0. Доколку може да се направи задолжителната тура, тогаш се проверува дали има преостанати средства за незадолжителните тури по редослед како што се наведени во листата. На крај се печати колку време ќе трае посетата на античките локалитети.

тури	Промет	Време	Задолжително
Хераклеа	1000	2 часа	ДА
Стоби	500	1 час	НЕ
Требенишко кале	1200	1.5 часа	НЕ

Решение:

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

```
int main(){
```

```
int pari;
float vreme=0.0;
printf("Vnesi kolku pari imash:");
scanf("%d",&pari);
if(pari>=1000){
vreme += 2.0;
pari-=1000;
if(pari>=500){
vreme+=1.0;
pari-=500;
}
if(pari>=1200){
vreme+=2.5;
pari -=1200;
}
}
printf("Turata ke trae %f vreme",vreme);
return 0;
}
```

Задача 2 – Група 1

Од тастатура се внесува број со парен број на цифри. Потоа внесениот број треба да се подели на 2 броја. Првиот број ќе биде формиран од вредностите на парните цифри на внесениот број прочитани во правилен редослед (од лево кон десно). Вториот број ќе биде формиран од вредностите на непарните цифри на внесениот број прочитани во правилен редослед (од лево кон десно). Потоа овие два броја треба да се помножат. Конечниот резултат е помножениот број прочитан во обратен редослед (односно, превртениот број). Да се испечати конечниот резултат. Доколку внесениот број има непарен број на цифри на екран да се печати "GRESKA" и да се прекине програмата.

Пример:

- Внесен број: 122556
- Прв број: 226
- Втор број: 155
- Помножен број: $226 * 155 = 35030$
- Конечен резултат: 3053

Решение:

```
#include <stdio.h>

int main(){

int broj,s1,s2,pom,proizvod,cifri,prv,vtor,cifra;
printf("Vnesi go brojot:");
scanf("%d",&broj);
```

```

cifri=0;
pom=broj;
while(pom){
    cifri++;
    pom/=10;
}
if(cifri%2==1){
    // neparen broj na cifri
    printf("GRESKA");
    return 0;
}
pom=broj;
prv=vtor=0;
s1=s2=1;
while(pom){
    cifra=pom%10;
    if(cifra%2==0){

        prv=cifra*s1+prv;
        s1*=10;
    }
    else {

        vtor =cifra*s2+vtor;
        s2*=10;}
    pom/=10;
}
proizvod=prv*vtor;
printf("prv = %d, vtor = %d\n",prv,vtor);
printf("Proizvodot e %d \n",proizvod);
pom=0;
while(proizvod){
    pom = pom *10 + (proizvod%10);
    proizvod/=10;
}
printf("konecniot rezultat e %d\n",pom);
return 0;
}

```

Задача 2 – Група 2

Од тастатура се внесува број со парен број на цифри. Потоа внесениот број треба да се подели на 2 броја. Првиот број ќе биде формиран од вредностите на парните цифри на внесениот број прочитани во обратен редослед (од десно кон лево). Вториот број ќе биде формиран од вредностите на непарните цифри на внесениот број прочитани во обратен редослед (од десно кон лево). Броевите да се испечатат. Потоа треба да се пресмета производот на цифрите од првиот број. Треба да се пресмета и производот на цифрите од

вториот број. Конечниот резултат е збирот од производот на цифрите од првиот број и производот на цифрите од вториот број. Да се испечати конечниот резултат. Доколку внесениот број има непарен број на цифри на екран да се печати "GRESKA" и да се прекине програмата.

Пример:

- Внесен број: 122556
- Прв број: 622
- Втор број: 551
- Производ од цифри на прв број: $6*2*2=24$
- Производ од цифри на втор број: $5*5*1=25$
- Конечен резултат: $24+25=49$

Решение:

```
#include <stdio.h>

int main(){
int broj,pom,suma,prv,vtor,cifra,cifri;
int p1,p2;
printf("Vnesi go brojot:");
scanf("%d",&broj);
cifri=0;
pom=broj;
while(pom){
    cifri++;
    pom/=10;
}
if(cifri%2==1){
// neparen broj na cifri
printf("GRESKA");
return 0;
}
prv=vtor=0;
pom=broj;
while(pom){
cifra=pom%10;
pom/=10;
if(cifra%2==0){
    prv = prv*10 + cifra;
} else {
    vtor = vtor*10 + cifra;
}
}
printf("prv = %d vtor = %d \n",prv,vtor);
pom=prv;
p1=1;
```

```
while(pom){
p1 = p1 * (pom%10);
pom/=10;
}
pom=vtor;
p2=1;
while(pom){
p2 = p2 * (pom%10);
pom/=10;
}
printf("rezultatot e %d\n",p1+p2);
return 0;
}
```

Задача 3 – Група 1

Да се напише функција која како аргумент добива природен број. Функцијата треба да провери дали секој пар соседни цифри во бројот го исполнуваат условот: “Ако на цифрата со помала тежина (односно десната цифра) се додаде 1, тогаш таа ќе биде делива (без остаток) со соседната цифра со поголема тежина (односно левата цифра).” Функцијата треба да врати вредност 1 ако бројот го исполнува условот за секој пар цифри. Во случај доколку барем еден пар цифри не го исполнуваат условот, функцијата треба да врати вредност 0.

Да се напише и главна програма која ќе ја тестира работата на функцијата.

Да се напише и **рекурзивна** верзија од функцијата.

Пример:

1. За $f(1235)$ -> функцијата треба да врати 1, бидејќи: за цифрата 5 имаме $5 + 1 = 6$ и 6 е деливо со три, за цифрата 3 имаме $3 + 1 = 4$ и 4 е деливо со два, за цифрата 2 имаме $2 + 1 = 3$ и 3 е деливо со 1.
2. За $f(1237)$ -> функцијата треба да врати 0, бидејќи: за цифрата 7 имаме $7 + 1 = 8$ и 8 не е деливо со три, останатите цифри може и не мора да се проверуваат (бидејќи секако имаме еден пар кој не го исполнува условот).

Решение:

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

```
int funk(int x){
```

```
int c1,c2;
```

```
while(x>9){
```

```
c1 = x%10;
```

```
x/=10;
```

```
c2 = x%10;
```

```
c1++;
```

```

if(c1%c2!=0){return 0;}
}
return 1;
}

int rek(int x){
    int c1,c2;
    if(x<10)return 1;
    c1 = x%10;
    c2 = (x/10)%10;
    c1++;
    if(c1%c2!=0){return 0;}
    return rek(x/10);
}

int main(){
    int broj;
    printf("Vnesi go brojot:");
    scanf("%d",&broj);
    if(rek(broj)==1){
        printf("Uslovot vazi\n");
    }else {
        printf("Uslovot ne vazi\n");
    }
    return 0;
}

```

Задача 3 – Група 2

Да се напише рекурзивна функција која за цел број, кој се пренесува како аргумент, ќе проверува дали барем еден од сите можни двоцифрени броеви, оформени од две соседни цифри на бројот, е парен број. Доколку е исполнет условот функцијата да враќа 1, во спротивно функцијата да враќа 0.

Да се напише и **рекурзивна** верзија од функцијата.

Пример: За бројот 23457 треба да се разгледаат следниве двоцифрени броеви: 23, 34, 45, 57 и бидејќи 34 е парен функцијата треба да врати вредност 1. За бројот 2357 треба да се разгледаат следниве двоцифрени броеви: 23, 35, 57 и бидејќи нема парен број функцијата треба да врати вредност 0.

Решение:

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

```
int funk(int x){
    int p;
    while(x>9){
        p = x%100;
        x/=10;
        if(p%2==0){return 1;}
    }
    return 0;
}
```

```
int rek(int x){
    if(x<10)return 0;
    int p = x%100;
    if(p%2==0)return 1;
    return rek(x/10);
}
int main(){
    int broj;
    printf("Vnesi go brojot:");
    scanf("%d",&broj);
    if(rek(broj)==1){
        printf("Uslovot vazi\n");
    }else {
        printf("Uslovot ne vazi\n");
    }
    return 0;
}
```

Задача 3 – Група 3

Да се напише рекурзивна функција која за цел број, кој се пренесува како аргумент, ќе проверува дали барем еден од сите можни двоцифрени броеви, оформени од две соседни цифри на бројот, е прост број. Доколку е исполнет условот функцијата да враќа 1, во спротивно функцијата да враќа 0.

Да се напише и **рекурзивна** верзија од функцијата.

Пример: За бројот 23457 треба да се разгледаат следниве двоцифрени броеви: 23, 34, 45, 57 и бидејќи 23 е прост функцијата треба да врати вредност 1. За бројот 2457 треба да се разгледаат следниве двоцифрени броеви: 24, 45, 57 и бидејќи нема прост број функцијата треба да врати вредност 0.

Решение:

```
#include <stdio.h>
int eprost(int x){
    int i;
    for(i=2;i<x;i++){
        if(x%i==0){return 0;}
    }
    return 1;
}
int funk(int x){
    int p;
    while(x>9){
        p = x%100;
        x/=10;
        if(eprost(p)==1){return 1;}
    }
    return 0;
}

int rek(int x){
    if(x<10)return 0;
    int p = x%100;
    if(eprost(p)==1)return 1;
    return rek(x/10);
}
int main(){
    int broj;
    printf("Vnesi go brojot:");
    scanf("%d",&broj);
    if(rek(broj)==1){
        printf("Uslovot vazi\n");
    }else {
        printf("Uslovot ne vazi\n");
    }
    return 0;
}
```