

ПРВ ПАРЦИЈАЛЕН ИСПИТ ПО
ПРОГРАМИРАЊЕ И АЛГОРИТМИ

23.11.2020

Задача 1 – Група 1

Да се напише програма во C програмскиот јазик за работа со список за рецепт за торта. Корисникот прво треба да внесе колку пари има на располагање, а потоа програмата треба, врз основа на табелата дадена подолу, да пресмета каква торта ќе може да се направи (според тоа колку пари има).

Производ	Цена	Задолжително
Брашно	30	ДА
Јајца	56	ДА
Шекер	32	ДА
Чоколадо	55	НЕ
Ванила	20	НЕ

Програмата треба прво да провери дали има вкупно пари да ги купи основните производи. Доколку има, тогаш се печати порака дека може да се направи торта, инаку, печати дека не може. Доколку може да се направи торта, проверува дали има пари за чоколадо и или ванила. Доколку има пари за чоколадо, тогаш се печати дека тортата ќе биде чоколадна, или доколку има пари за ванила, тогаш се печати дека тортата ќе биде со вкус на ванила. Доколку има пари и за двете, тогаш се печати дека тортата ќе биде чоко-ванила.

Решение:

```
#include <stdio.h>

int main(){
    int pari = 0;

    int brasno = 30, jajca=56, sheker=32, cokolado=55, vanila=20;

    printf("Vnesi kolku pari imas na raspolaganje\n");

    scanf("%d",&pari);

    if(pari>=(brasno+jajca+sheker)){
        printf("Imate dovolno pari za da napravite torta\n");
        pari-=(brasno+jajca+sheker);
    }
    if(pari>=(cokolado+vanila)){
```

```
printf("Tortata kje bide coko-vanila\n");

pari-= (cokolado+vanila);
}else if (pari>=cokolado){

printf("Tortata kje bide cokoladna\n");

pari -= cokolado;
}else if(pari>=vanila){

printf("Tortata kje bide so vkus na vanila\n");

pari-=vanila;
}

}else {

printf("Nemate dovolno pari za da napravite torta\n");

}

// pecatenjeto kolku pari vi oстанаа не е барање од zadacata

printf("Vi preostanaа %d pari\n",pari);

return 0;

}
```

Задача 2 – група 2

Да се напише програма во C програмскиот јазик за достава за пица. Корисникот прво треба да внесе колку време има преостанато до завршување на смената, а потоа програмата треба, врз основа на табелата дадена подолу, да пресмета колкав промет ќе оствари од доставата на пици во преостанатиот временски период.

Достава	Промет	Време	Задолжително
Центар	300	10 мин	ДА
Аеродром	700	25 мин	ДА
Карпош	1020	15 мин	ДА
Ченто	450	35 мин	НЕ
Ѓорче	500	42 мин	НЕ

Програмата треба прво да провери дали има вкупно време да се достават сите задолжителни доставки. Доколку можат сите задолжителни доставки да се направат, тогаш се зема во предвид прометот од тие доставки, и се проверува дали може да се достават и дополнителните доставки. Инаку, не се прави ниту една достава и прометот има вредност 0.

Доколку може да се достават задолжителните, тогаш се проверува за дополнителните по редослед како што се наведени во листата. На крај се печати пресметаниот промет.

Решение:

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

```
int main(){
int vreme = 0,promet=0;
int centarPromet=300, centarVreme=10, aerodromPromet=700, aerodromVreme=25,
karposPromet=1020, karposVreme=15, centoPromet=450, centoVreme=35,
gjorcePromet=500,gjorceVreme=42;
printf("Vnesi kolku imas preostanato vreme do kraj na smena\n");
scanf("%d",&vreme);
if(vreme>=(centarVreme+aerodromVreme+karposVreme)){
printf("Imate vreme za da napravite osnovna dostava\n");
promet += (centarPromet+aerodromPromet+karposPromet);
vreme -= (centarVreme+aerodromVreme+karposVreme);
if(vreme >=centoVreme){
printf("Imate vreme za da napravite dostava i do chento\n");
promet += centoPromet;
vreme-=centoVreme;
}
if(vreme>=gjorceVreme){
printf("Imate vreme za da napravite dostava i do gjorce\n");
promet += gjorcePromet;
vreme-=gjorceVreme;
}
} else {
printf("Nemate vreme za da napravite osnovna dostava\n");
}
printf("Prometot e %d\n",promet);
return 0;
}
```

Задача 2 – Група 1

Да се напише програма што ќе ги најде и отпечати броевите помали од n кои што се “интересни”. За интересен број се смета бројот чиј обратен број е делив со бројот на неговите цифри.

Доколку не се пронајде таков број, да се отпечати соодветна порака на екран.

Пример: 356 => 653 е преврteniот broj i ne e deliv so 3

216 => 612 e prevrteniот broj i e deliv so 3

Решение:

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

```
int main() {
    int n, pom, prev=0, cifra, brcif=0, i;

    printf("Vnesi go n\n");
    scanf("%d", &n);
    for(i=1; i<n; i++){
        pom = i;
        prev=0;
        brcif=0;
        while(pom){
            cifra = pom%10;
            pom/=10;
            prev = prev*10 + cifra;
            brcif++;
        }
        if(prev%brcif==0){
            printf("Prevrteniot broj e %d i e deliv so %d\n", prev, brcif);
        } else {
            printf("Prevrteniot broj e %d i ne e deliv so %d\n", prev, brcif);
        }
    }
    return 0;
}
```

Задача 2 – Група 2

Да се напише програма што ќе ги најде и испечати броевите помали од n кои што се “интересни”.

За интересен број се смета бројот чиј обратен број при делење со производот на неговите цифри дава остаток 2.

Доколку не се пронајде таков број, да се испечати соодветна порака на екран.

Пример: $311 \Rightarrow 113$ е обратниот број и при делење со 3 дава остаток 2

Решение:

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

```
int main() {
    int n, pom, prev=0, cifra, prcif=1, i;
```

```
printf("Vnesi go n\n");
scanf("%d",&n);
for(i=1;i<n;i++){
    pom = i;
    prev=0;
    prcif=1;
    while(pom){
        cifra = pom%10;
        pom/=10;
        prev = prev*10 + cifra;
        prcif*=cifra;
    }
    if(prcif!=0&&prev%prcif==2){
        printf("Prevrteniот број е %d i pri delenje so %d dava ostatok 2\n",prev,prcif);
    } else {
        printf("Prevrteniот број е %d i pri delenje so %d ne dava ostatok 2\n",prev,prcif);
    }
}
return 0;
}
```

Задача 3 – Група 1

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.

Забелешка: Ако функцијата е рекурзивна и задачата е точна, задачата носи 35 поени. Доколку задачата ја решавате без рекурзија, максималниот број на поени кој може да го освоите е 25 поени (доколку ви е потполно точна).

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: решението вклучува рекурзивна и нерекурзивна функција за истото решение

```
#include <stdio.h>
int rek(int b){
    int broj;
    if(b< 10){
        // brojot e ednocifren, izminati se site cifri i nemame sretnato nitu eden dvocifren sto e paren,
        vrati 0
        return 0;
    }
}
```

```

broj = b%100;
if(broj%2==0){
    // dvocifreniot broj e paren, vrati 1
    return 1;
}
return rek(b/10);
}
int nerek(int b){
int broj;
while(b>9){
    broj = b%100;
    if(broj%2==0){return 1;}
    b/=10;
}
return 0;
}
int main(){
int broj;
printf("Vnesi broj\n");
scanf("%d",&broj);
if(nerek(broj)==1){ // if(rek(broj)==1){
    printf("Brojot go ispolnuva uslovot");
} else {
    printf("Brojot ne go ispolnuva uslovot");
}
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 прост број. Доколку е исполнет условот функцијата да враќа 1, во спротивно функцијата да враќа 0.

Забелешка: Ако функцијата е рекурзивна и задачата е точна, задачата носи 35 поени. Доколку задачата ја решавате без рекурзија, максималниот број на поени кој може да го освоите е 25 поени (доколку ви е потполно точна).

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: решението вклучува рекурзивна и нерекурзивна функција за истото решение

```
#include <stdio.h>
int eprost(int b){
    int i;
    for(i=2;i<b/2;i++){
        if(b%i==0)return 0;
    }
    return 1;
}
int rek(int b){
    int broj;
    if(b< 10){
        // brojot e ednocifren, izminati se site cifri i nemame sretnato nitu eden dvocifren sto e paren,
        vrati 0
        return 0;
    }
    broj = b%100;
    if(eprost(broj)==1){
        // dvocifreniot broj e prost, vrati 1
        return 1;
    }
    return rek(b/10);
}
int nerek(int b){
    int broj;
    while(b>9){
        broj = b%100;
        if(eprost(broj)==1){return 1;}
        b/=10;
    }
    return 0;
}
int main(){
    int broj;
    printf("Vnesi broj\n");
    scanf("%d",&broj);
    if(nerek(broj)==1){ // if(rek(broj)==1){
        printf("Brojot go ispolnuva uslovot");
    }else {
        printf("Brojot ne go ispolnuva uslovot");
    }
    return 0;
}
```