

Examenul de bacalaureat național 2020
Proba E. d)
Informatică
Limbajul C/C++

Testul 13

Filieră teoretică, profil real, specializare științe ale naturii

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

I. TÉTEL (20 punct)

Minden item esetében 1-től 5-ig írja a vizsgalapra a helyes válasznak megfelelő betűt. Minden helyes válasz 4 pontot ér.

- Adjátok meg azt a C/C++ kifejezést, amelynek akkor és csakis akkor 1 az értéke, ha az n természetes szám típusú változóban tárolt érték osztható 4-gyel és 5-tel is.
 - $!(n\%4!=0 \ || \ n\%5!=0)$
 - $!(n/4==1 \ || \ n/5!=0)$
 - $n\%4==0 \ \&\& \ !(n\%5==0)$
 - $n/4==0 \ \&\& \ !(n/5==0)$
- Annak megállapítására, hogy egy egydimenziós tömbben létezik-e az $x=4$ értékű elem, a bináris keresés módszerét használjuk. Az elemek, amelyekkel az x értékét összehasonlítjuk a módszer alkalmazása során a következők: 14, 8, 4.
A tömb elemei lehetnek:
 - (48, 14, 9, 8, 7, 4, 2)
 - (14, 14, 8, 8, 4, 4)
 - (14, 8, 4, 3, 2, 0)
 - (4, 8, 9, 14, 16, 24, 48)
- Az A és B egydimenziós tömbök elemei: $A=(50, 36, 27, 20, 2)$, és $B=(63, 45, 8, 5, 3)$. Ezek növekvő sorrendbe való összefésülésének következtében létrejött tömb elemei:
 - (2, 3, 5, 8, 20, 27, 36, 45, 50, 63)
 - (2, 5, 8, 27, 50)
 - (50, 36, 27, 20, 2, 63, 45, 8, 5, 3)
 - (50, 63, 36, 45, 8, 27, 5, 20, 2, 3)
- Egy olyan C/C++ kifejezés, amelynek értéke 1.0:
 - $\text{ceil}(-0.2080)$
 - $\text{ceil}(0.8020)$
 - $\text{floor}(-0.2080)$
 - $\text{floor}(0.8020)$
- A mellékelt utasításokban, minden változó egész típusú. A kódrészlet végrehajtása után az x változó értéke a 2020 és 2200 értékek legnagyobb közös osztója lesz, ha a pontozott rész helyett beírt kifejezés a következő:

$x=2020; y=2200;$ $\text{while} (y!=0)$ $\{ z=...; x=y; y=z; \}$
--

 - $x\%2$
 - $x/2$
 - $x\%y$
 - x/y

II. TÉTEL (40 pont)

- Adott a mellékelt algoritmus pszeudokódban.
Az $a\%b$ jelöli az a természetes szám b nem nulla természetes számmal való osztási maradékát, és $[c]$ a c valós szám egész részét.
 - Írjátok le az értéket, amely kiíródik az algoritmus végrehajtása-kor, ha a beolvasott számok sorrendben 12345, 25, 2070, 49, 270135, 21. (6p.)
 - Ha az n -be beolvasott érték 100, írjátok le egy adatsort a $[10^3, 10^4)$ intervallumból, amelyet folytatólagosan beolvasva, az algoritmus által kiírt eredmény 10. (6p.)
 - Írjátok az adott algoritmusnak megfelelő C/C++ programot. (10p.)
 - Írjátok pszeudokódban az eredetivel egyenértékű algoritmust, amely csak egyetlen ismétlődő struktúrát tartalmaz. (6p.)

Beolvas n (természetes szám) $p \leftarrow 1; m \leftarrow 0; k \leftarrow 0$ amíg $n \neq 0$ végezd el beolvas x (természetes szám) $i \leftarrow k$ amíg $i \neq 0$ végezd el $x \leftarrow [x/10]; i \leftarrow i-1$ ■ ha $x=0$ akkor $c \leftarrow n\%10$ különben $c \leftarrow x\%10$ ■ $m \leftarrow c*p+m; n \leftarrow [n/10]$ $p \leftarrow p*10; k \leftarrow k+1$ ■ kiír m
--

2. A **raza** egész típusú változó egy kör sugarát tárolja, míg a **centruX** és **centruY** valós változók egy kör középpontjának az **xOy** síkban megadott, valós koordinátáit (abszcissza és ordináta) tárolják. Értelmezétek megfelelőképpen a **centruX** és **centruY** változókat, majd írjatok egy utasítássort, amelynek eredményeként a képernyőre az **Unitar** üzenet kerül, ha a megadott kör középpontja a koordináta rendszer origójában van és sugara 1, vagy az **Altceva** üzenetet, ellenkező esetben. **(6p.)**
3. Az **i** egész típusú változó, míg a többiek **char** típusúak. Írjátok le, mit ír ki a mellékelt kód végrehajtása során. **(6p.)**
- ```
c1='s';c2='o';c3='u';
cout<<c2<<c3<<endl; | printf("%c%c\n",c2,c3);
for(i=0;i<4;i++)
{ c=c1+(i-1)*(1-i%2)+3*(2*i/3-1)*(i%2);
 if(i==0) c1=c;
 cout<<c; | printf("%c",c);
}
```

### III. TÉTEL

**(30 pont)**

1. Beolvasunk egy  $n$  ( $n \geq 2$ ) természetes számot, és kiírjuk, ebben a sorrendben, a  $d$  és  $p$  számokat, ahol a  $d$  az a prímosztó az  $n$  szám prímtényezőre bontásában, amely a legkisebb  $p$  hatványon jelenik meg. Ha több ilyen prímosztó van, a legkisebbet kell kiírni. Adjátok meg pszeudokódban a leírt feladat megoldását. **Példa:** ha  $n=10780$ , akkor a kiírt értékek  $5 \ 1$  ( $10780=2^2 \cdot 5 \cdot 7^2 \cdot 11$ ). **(10p.)**
2. Írjatok egy C/C++ programot, amely a billentyűzetről beolvas két természetes számot,  $n-t$  és  $k-t$ , a  $[2, 20]$  intervallumból, majd felépít a memóriában egy egydimenziós tömböt  $n \cdot k$  darab elemmel, a következőképpen: a tömb elemei növekvő sorrendben vannak, az első tag az  $n$ , minden érték pontosan  $k$  alkalommal jelenik meg a sorban, és bármely két egymás melletti tag értékei egyenlőek, vagy egymást követő értékűek. A program írja ki a képernyőre a felépített tömböt, az elemeket egy-egy szóközzel elválasztva. **Példa:** ha  $n=3$  és  $k=4$ , az előállított tömb: **(10p.)**
- ```
3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6
```
3. Az $1, 2, 4, 8, 16, 32, \dots$ sorozat matematikai leírása a következő: $f_1=1, f_n=2 \cdot f_{n-1}$ (ahol az n egy természetes szám $n \geq 2$) A billentyűzetről beolvasunk egy x ($x \leq 10^9$) természetes számot, a megadott sorozat egyik tagjának értéke. A **bac.txt** állományba csökkenő sorrendbe, egy-egy szóközzel elválasztva, írassuk ki a sorozat minden olyan tagját, amelyek kisebbek vagy egyenlőek x -szel. Tervezzetek egy a memóriahasználat és végrehajtási idő szempontjából hatékony algoritmust. **Példa:** ha a beolvasott szám 16 , a **bac.txt** tartalmazni fogja a $16 \ 8 \ 4 \ 2 \ 1$ számokat.
- a. Írjátok le saját szavaitokkal a használt algoritmust, és indokoljátok annak hatékonyságát. **(2p.)**
- b. Írjátok a tervezett algoritmusnak megfelelő C/C++ programot. **(8p.)**