Babeș-Bolyai Tudományegyetem
FIZIKA KAR

Családnév: _____

LICENSZVIZSGA – 2021. június 29.

Keresztnév: _____

FIZIKA INFORMATIKA szak

1. próba: Alap- és szakismeretek értékelése

Feleletválasztós teszt

Kérjük, karikázza be az alábbi kérdéseknél az egyetlen helyes választ!

1. Az Ising modell Metropolis Monte Carlo szimulációjában a vizsgált rendszer hőmérsékeltét
   (a) a Metropolis algoritmus tartalmazza.
   (b) a kölcsönhatási energia kifejezése tartalmazza.
   (c) elhanyagoljuk.
2. A Kuramoto modellben a fázisátalakulás tanulmányozására
   (a) a mágnesezettség rendparamétert használtuk.
   (b) az átlaghőmérséklet rendparamétert használjuk.
   (c) az $r = \frac{1}{n}\sum_i(\cos\theta_i + \mathrm{i}\cdot\sin\theta_i)$ rendparamétert használjuk.
3. Gázak molekuláris dinamika szimulációjában a makroszkópikus mennyiségek
   (a) sokaságátlagát számoljuk ki.
   (b) időátlagát számoljuk ki.
   (c) sokaság és időátlagát számoljuk ki.
4. Jelöljük meg a numerikus gyökkereső módszerekre vonatkozó egyedüli helyes kijelentést.
   (a) A felező módszer lassabban konvergál mint az iterációs módszer.
   (b) Az érintő módszer lassabban konvergál mint az iterációs módszer.
   (c) A húr módszer robusztusabb mint a felező módszer.
5. Egy mechanikai rendszerre vonatkozó Newtoni-mozgástörvényben megjelenő valamely paraméter nem pontos. Milyen típusú hibát eredményez az említett bizonytalanság az egyenlet megoldásában, feltételezve, hogy Runge-Kutta módszert használunk?
   (a) kerekítési hibát
   (b) képlet hibát
   (c) öröklött hibát
6. Az alábbi, numerikus integrálásra vonatkozó állítások közül melyik hamis?
   (a) A Simpson-módszer hibája kisebb mint a trapéz módszeré
   (b) A kiterjesztett kvadratúra képleteket improprius integrálokra alkalmazzuk
   (c) A Romberg-integrál egyenlőközű felosztást alkalmaz
7. A Hall állandó előjele megegyezik
   (a) az elektronok elektromos töltésének előjelével.
   (b) a lyukak elektromos töltésének előjelével.
   (c) a többségi töltéshordozók előjelével.

8. Tiszta (intrinsic) félvezető esetén a Fermi nívó növekszik a hőmérséklet növekedésével ha

   (a) $m^*_{val} = m^*_{vez} \mathrm{m}^*_{val} > m^*_{vez}$

(b) $m^*_{val} < m^*_{vez}$

9. Ha két különböző Fermi nívóval $(E_{F_1} > E_{F_2}) rendelkez fl vezetkristlytsszerintmk, akkorahatr felletenelektronramotf$

10. kétirányú.

11. az 1-es félvezetőből a 2-esbe halad.

12. a 2-es félvezetőből az 1-esbe halad.

A Pauli-féle kizárási elv

1. megtiltja, hogy egynél több elektron ugyanazon a helyen legyen
2. megtiltja, hogy egynél több elektron ugyanabban a kvantumállapotban legyen
3. szerint legtöbb két elektron lehet ugyanabban a kvantumállapotban

A fényelektromos hatás esetén a zárófeszültség

1. A katód anyagától függ
2. Csökken a beérkező fotonok frekvenciájának növekedésével
3. Csak akkor határozható meg, ha a fotonok frekvenciája kisebb a küszöbfrekvenciánál

A héliumatom esetén

1. Megengedett az optikai átmenet a parahélium és az ortohélium között
2. A ^3P energiaszint három finomszerkezeti szintre hasad fel
3. Az alapállapot termje $^3S_1$

Adott az N elemű X sorozat, amely egész számokat tárol. Szeretnénk eldönteni, hogy minden eleme nagyobb-e, mint az adott B szám. Válasszák ki az alábbi algoritmusok közül azt, amely igaz értéket térít, ha a tulajdonság teljesül és hamis értéket különben.

1.
```
Algoritmus Döntés_2(N, X, B):
    i ← 1
    talált ← hamis
    Amíg nem talált és i ≤ N végezd el:
        Ha nem Xi <= B akkor
            i ← i + 1
        különben
            talált ← igaz
        vége(ha)
    vége(amíg)
    térítsd talált
Vége(algoritmus)
```

2.
```
Algoritmus Döntés (N, X, B):
    i ← 1
    Amíg i ≤ N és nem Xi > B végezd el:
        i ← i +| 1
    vége(amíg)
    térítsd i ≤ N
Vége(algoritmus)
```

3.
```
Algoritmus Döntés (N, X, B):
    i ← 1
    Amíg i ≤ N és Xi > B végezd el:
        i ← i + 1
    vége(amíg)
    térítsd i > N
Vége(algoritmus)
```

Melyik algoritmus rendezi helyesen az n elemű x sorozatot növekvő sorrendbe?

1.
```
void Rendezes(int n, int x[]){
   bool rendezett;
   int k = n;
   do{
      rendezett = true;
      int nn = k;
      for(int i = 1; i <= nn; i++){
         if(x[i] > x[i + 1]){
            csere(x[i], x[i + 1]);// felcseréli x[i]-t x[i + 1]-gyel
            rendezett = false;
            k = i;
         }
      }
   }while(rendezett);
}
```

2.
```
void Rendezes(int n, int x[]) {
   for(int j = 2; j <= n; j++){
      int seged = x[j];
      int i = j - 1;
      while(i > 0 && x[i] > seged){
         x[i + 1] = x[i];
         i--;
      }
      x[i + 1] = seged;
   }
}
```

3.
```
void Rendezes(int n, int x[]){
   int k = maximum(n, x);// téríti az n elemű x sorozat maximumát
   int seged[100];
   for(int i = 1; i <= k; i++)
      seged[i] = 0;
   for(int j = 1; j <= n; j++)
      seged[x[j]]++;
   int q = 0;
   for(int i = 1; i <= n; i++){
      for(int j = 1; j <= k; j++){
         q++;
         x[q] = i;
      }
   }
}
```

Válasszák ki a helyes választ:

1. A ”programozási tétel„ egy algoritmus-minta, amely garantáltan helyes és optimális megoldást ad egy bizonyos feladatosztály esetében.

2. A ”programozási tétel„ egy matematikai állítás, amely garantáltan helyes és optimális megoldást ad egy bizonyos feladat esetében.
3. A ”programozási tétel„ megadja matematikai állítások formájában egy bizonyos feladat megoldását.