

Jelfeldolgozás a közlekedésben, II. ZH

NÉV:

2015. május 14. 12:15-14:00, ST121 NEPTUN KÓD:

A. Készítsen C programot, amely a potenciométer segítségével egy V43-as sorozatú mozdony controllerét szimulálja, és jelzi a virtuális kijelzőn a következőképpen:

- Ha a P4.4 kapcsoló ON, akkor a mozdony ELŐRE irányban halad, a kijelzőn az ELORE és a FOKOZAT: érték (0%-100%) jelenik meg – tetszőleges helyen.
- Ha a P4.4 kapcsoló OFF, akkor a mozdony HÁTRA irányban halad, a kijelzőn a HATRA és a FOKOZAT: érték (0%-100%) jelenik meg – tetszőleges helyen.
- A P4.4 kapcsoló átkapcsolása csak akkor lehetséges, ha a FOKOZAT érték 0%. (Vagyis menet közbeni átkapcsolásnál az irány nem változik meg, az csak akkor fog, ha letekerjük a kontrollert, és a következő feltekerésnél vesszük csak figyelembe.)
- A jármű álló helyzetből indul, azaz bekapcsolás után csak akkor van irány, ha a menetkapcsolón a fokozat 0!
- 3 másodpercenként a NYUGTA felirat jelenjen meg a kijelzőn, amelyet 3 másodperc alatt nyugtázni kell a P5.0 gomb lenyomásával. Ha nem történik nyugtázás, az irány eltűnik (vérszéfékezés) – éberségi funkció. Vérszéfékezés után a jármű újraindul, ha a P5.0-ás gombot megnyomtuk!



Jelfeldolgozás a közlekedésben, II. ZH

NÉV:

2015. május 14. 12:15-14:00, ST121 NEPTUN KÓD:

B. Készítsen C programot, amely egy jármű kijelzőjét szimulálja a virtuális kijelzőn a következőképpen:

- A program elején válasszuk ki a járatszámot és a végállomást (tetszőleges, de férjen ki az első sorba, azaz max. 16 karakter!)) Ehhez definiáljunk előre 3 lehetséges változatot, amelyeket sorban 3 másodpercenként – egymás után - kiírunk a kijelző felső sorába, és ha valamelyik esetén lenyomjuk a P5.3-at, akkor az rögzül az első sorban!
- A rögzítés után a jármű elindul, és az útvonalán 4 megállója van (ebből a 4. a végállomás), mindegyiknek van neve (max. 16 karakter) és átszállási lehetőségei (max. 16 karakter). A megálló nevét és a hozzátartozó átszállási lehetőségeket 4 másodperces váltásban írja ki a kijelző második sorára. (Rögzítés után automatikusan az első megálló és átszállási kapcsolatai kerülnek kijelzésre.)
- A következő megállóra ugrás a KeyPad A gombjának lenyomásával (ha van ilyen), míg az előző megállóra ugrás a KeyPad B gombjának lenyomásával (ha van előző megálló) történjen!
- Egy gombnyomásra csak egyet lépjen tovább a program!
- A KeyPad D gombjának lenyomása után kezdődjön előlről a program, azaz kérje a viszonylat kiválasztását!



Jelfeldolgozás a közlekedésben, II. PZH

NÉV:

2015. május 21. 12:00-14:00, ST121 NEPTUN KÓD:

A. Készítsen C programot, amely a KeyPad segítségével egy Tatra T5C5K sorozatú villamos ajtóvezérlést szimulálja, és jelzi a virtuális kijelzőn a következőképpen:

- A jármű álló helyzetből indul, és az összes ajtó zárva van. A kijelzőn az AJTOK ZARVA felirat látható (tetszőleges helyen).
- Az utasok a KeyPad 1-6 nyomógombokkal jelezhetik a leszállási szándékukat. Ha a P4.2 kapcsoló ON, akkor a bal, míg ha ez a kapcsoló OFF, akkor a jobb oldali ajtók fognak kinyílni. A KeyPad A gombjának megnyomására az ajtók kinyílnak, és a kijelzőn a BAL OLALI AJTOK NYITVA felirat vagy a JOBB OLDALI AJTOK NYITVA felirat jelenik meg (figyelem, ezek nem fog kiférni egy sorba!), Ha nem volt leszállás jelzés, az ajtók nem nyílnak ki.
- Az ajtók zárása a következőképpen történik: először a KeyPad B gombjának lenyomására megszólal a csengő (felirat: CSENGO), majd a C gomb lenyomására az ajtók bezáródnak, és ismét az AJTOK ZARVA felirat jelenik meg.
- A KeyPad D gombjának lenyomására csak a jobb első ajtó nyílik ki, ekkor a felirat: JOBB ELSO NYITVA felirat jelenik meg. A jobb első ajtó zárása a # gomb lenyomására történik, ekkor egy 4 másodperces időzítő lejártá után az ajtó automatikusan bezáródik.



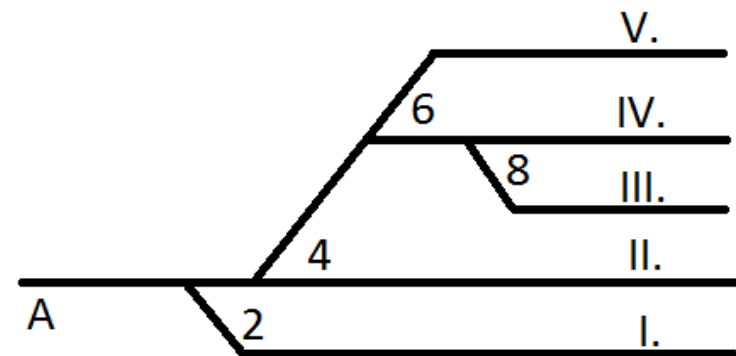
Jelfeldolgozás a közlekedésben, II. PZH

NÉV:

2015. május 21. 12:00-14:00, ST121 NEPTUN KÓD:

A. Készítsen C programot, amely a KeyPad segítségével egy egyszerűsített állomást szimulál, és jelzi a virtuális kijelzőn a következőképpen:

- Az állomás páros végén 4 kitérő található (lásd ábra), melyek közül a 2-es és a 4-es kitérő irányba 80 km/h-val járható, alapállapotuk (bekapcsolás után) egyenes. A váltó állások (E vagy K) a kijelző első sorában vannak kijelvezve folyamatosan a következőképpen (pl.): 2:E 4:K 6:K 8:E. A kijelző második sorában az A jelző állapota látható, alaphelyzetben: A:V (V mint, vörös).
- A váltók a KeyPad 2, 4, 6 és 8 gombjával vezérelhetők kitérő/egyenes állásba.
- A KeyPad A gombjának lenyomására felépül az A bejárat jelzőtől a kitérő állapotok alapján lehetséges vágányút, és 3 másodperccel a gomb lenyomása után az A bejárat jelzőn megjelenik a szabad fény - MÁV F1F2 alapján - (feltételezzük, hogy valamennyi kijárat jelzőn vörös fény világít). A jelzőállapotát a kijelző alsó sorába a következőképpen írjuk ki – pl, ha a IV. vg-ra állítható be a vágányút, akkor – A: S1 S2 Zi, ahol Zi a zöld indikátor.
- Szabad jelzési kép esetén (az A gomb lenyomása után) a váltók nem állíthatóak.
- A pontszerű járműérzékelés hatására (B nyomógomb lenyomása) a vágányút feloldódik, az A bejárat jelző ismét vörös lesz, és a váltók állíthatóak lesznek.



Jelfeldolgozás a közlekedésben, II. ZH - 2016

NÉV:

2015. május 19. 12:15-14:00, ST122 NEPTUN KÓD:

A. Készítsen C programot, amely egy futópad kijelzőjét szimulálja a következőképpen:

- A program a virtuális kijelző második sorában az potencióméterrel beállított nehézségi szintet jelenti meg, ahol összesen 16 lehetséges fokozat van. Minden egyes fokozatot a fokozat számának megfelelő darabszámú „/” karakter jelent (karakterkód: decimális 47). A kijelző felső sorának első 6 karaktere fixen: „SZINT:”. A kijelzés a második sor bal széléről indul, azaz ha az első fokozatban vagyunk, akkor az alsó sor első karaktere: /, ha a másodikban, akkor az első 2 karakter: /, stb.
- A futóédzés a KeyPad „A” gombjának lenyomására indul el, és futás közben az elégetett kalória mennyiségét írja ki az első sor utolsó 7 karakterére. A kiírandó hét karakter első 4 értéke az elégetett kalória xxxx formátumban (0-9999 közötti érték). Az utolsó 3 karakter a „CAL” felirat. Az elégetett kalóriaértéket 0,5 másodpercenként írja ki a program, és a kiírandó érték = előző érték+aktuális fokozat. A futóédzés csak akkor indulhat el, ha a fokozat=0!
- A KeyPad „B” gombjának lenyomásával a futóédzés véget ér, és a kijelzőn az utoljára kiírt érték rögzül.
- A „reset” gomb lenyomására a program alapállapotba kerül, várjuk az újbóli indítást.



A kép illusztráció!

Jelfeldolgozás a közlekedésben, II. ZH - 2016

NÉV:

2015. május 19. 12:15-14:00, ST122 NEPTUN KÓD:

B. Készítsen C programot, amely a KeyPad segítségével egy lift vezérlőpaneljét szimulálja a következőképpen:

- A lift egy 3 emeletes épületben található, a szintek: földszint, 1. emelet, 2. emelet, 3. emelet. Az aktuális szint az első sorban kerül kijelzésre.
- A lift alapértelmezetten a földszinten van, az ajtók nyitva. Ekkor a második sorban az „AJTÓK NYITVA” felirat szerepel. (Az első sorban természetesen a földszint felirat jelenik meg.)
- Az ajtók a 0,1,2 vagy 3 jelű gomb megnyomására 3 másodperc alatt záródnak be, a zárás alatt a kijelző alsó sorában az „AJTÓK ZÁRÓDNAK” kijelzés szerepel, majd a lift elindul. Ha a lift menet közben van, akkor az első sorban a „MENET KÖZBEN” felírra vált a kijelző.
- A lift a kívánt emeletre az emeletek közötti távolság*3 másodperc alatt ér oda - pl. a földszint - 3. emelet közötti távolság 3*3 másodperc (azonos emeletek között nincs menet), majd újabb 3 másodperc kell az ajtók nyitásához, mialatt az „AJTÓK NYÍLNAK” felirat szerepel a második sorban. Ekkor már az első sorban az új aktuális szint kerül kijelzésre.
- Ha az ajtók kinyíltak, ismét megjelenik az ajtók nyitva felirat, és várjuk a következő menetet.
- A kiírandó karaktertömböket `const char valtozonev[]=" ";` -ként deklaráljuk!!!! (pl: `const char elso[]="1. emelet";`)



A kép illusztráció!

Jelfeldolgozás a közlekedésben, II. PZH - 2016

NÉV:

2016. május 24. 16:00-18:00, ST122 NEPTUN KÓD:

PZH. Készítsen C programot, amely a KeyPad segítségével egy jegykiadó automatát szimulál a következőképpen:

- A jegykiadó automata csak vonaljegyvet árusít, amelynek ára 350 Ft, az automata pénzt visszaadni nem tud, és csak **10, 20, 50 és 100 FT-os érmét** fogad el.
- Alapállapotban a kijelezőn a következő üzenet látható: 'Kezdeshez nyomja meg a D gombot'!
- A D nyomógomb megnyomása után az első sorban a 'Bedobott összeg:' felirat kerül kijelzésre, a második sorban pedig kijelzésre kerül a bedobott pénzösszeg 'xxxFT' formátumban.
- A KeyPad 0 gombjának megnyomása 100 Ft-al, az 1-es 10 Ft-al, a 20-as 20 Ft-al, míg az 5-ös 50 Ft-al növeli a már bedobott pénzösszeget, az eddig bedobott összeg folyamatosan megjelenik a kijelzőn. (Egy gomb megnyomására csak egyet számolunk!)
- Ha pontosan bedobtunk 350 Ft-ot, akkor a kijelzőn 5 másodpercig a „nyomtatas” felirat látszódik, majd ez után újabb 5 másodpercig a 'Szep napot, viszontlatasra' felirat látszódik. Az 5 másodperc lejártá után az automata alapállapotba kerül, várjuk az újbóli vásárlást.
- A kiírandó karaktertömböket const char valtozonev[]=" ”; -ként deklaráljuk!!!! (pl: const char nyomt[]="nyomtatas”;



A kép illusztráció!

Mechatronika és mikroszámítógépek, II. ZH - 2016.

NÉV:

2016. december 7. 12:15-14:00, ST122 NEPTUN KÓD:

ZH/A. Készítsen C programot, amely egy konténerakkodó bakdaru működését szimulálja a következőképpen:

- A daru álló helyzetből indul, és a virtuális kijelző első sorában a „A daru áll!” üzenet olvasható. Indításkor az emelő szerkezet felső helyzetben van, azaz a potméter a jobb szélső állapotban van. Ez a második sorban kerül kijelzésre a következőképpen: „Emelő fent”. Ha az emelő indításkor nincs fent, akkor a program csak akkor lép majd tovább, ha a potmétert a megfelelő állásba tekertük, és csak ekkor jelenik meg az emelőre vonatkozó üzenet.
- A daru előre vagy hátrafelé tud mozogni (ha az emelő fent van), mégpedig előre a numpad A, hátra pedig a numpad D gombjának folyamatos nyomva tartásával. Ekkor az első sorban az „ELŐRE” vagy „HÁTRA” üzenet olvasható. (Ha a daru nem mozog, az első sorban „A daru áll!” üzenet olvasható.
- Ha a daru áll, lehetőség van az emelő szerkezet mozgatására a potméter segítségével. Ekkor az „Emelő mozgásban” üzenet látható. Ha a potméter végállásban van, akkor az „Emelő lent” vagy „Emelő fent” üzenet olvasható a második sorban. (Ha a daru mozog, a potméter tekerésének nincs hatása.)
- A daru működtetése veszélyes, ezért folyamatosan meg kell győződni arról, hogy a kezelő ébren van. Ez azt jelenti, hogy 10 másodpercenként ki kell írni a második sor utolsó karakterhelyére egy nagy F betűt. A kiírás után újabb 10 másodperc áll rendelkezésre arra, hogy a kezelő a P5.1 vagy P5.2 gombok megnyomásával nyugtázza a felhívást. Ha a kezelő időben nyugtázza a felhívást, nem történik semmi sem.
- Ha a kezelő nem nyugtázza időben az éberségi felhívást, akkor a program leáll, a kijelzőn „SYSTEM FAILURE” üzenet látható, és a program csak reset hatására indul újra.



A kép illusztráció!

Mechatronika és mikroszámítógépek, II. ZH - 2016.

NÉV:

2016. december 7. 12:15-14:00, ST122 NEPTUN KÓD:

ZH/B. Készítsen C programot, amely a potenciométer segítségével egy játékot szimulál a következőképpen:

- A program elején írjuk ki a kijelzőre, hogy: Kezdéshez nyomja meg a C gombot!
- A C gomb lenyomása után a program beolvassa és eltárolja a potenciométerrel beállított értéket (0-255 közötti szám). A beolvasás után a felhasználónak 30 másodperce van kitalálni, a P4-es kapcsolók megfelelő beállításával a potméterről beolvasott értéket.
- A P4-es porton a P4.0 a 2^0 , a P4.1 a 2^1 , a P4.2 a 2^2 (stb) értéknek felel meg, ha a kapcsoló állapota ON, azaz az a cél, hogy a kapcsolók megfelelő beállításával megtaláljuk az aktuálisan a potméterről beolvasott értéket.
- A kijelzőn folyamatosan írjuk ki, hogy a P4-es porton beállított kombináció a keresett számnál kisebb vagy nagyobb, és a P4-es port aktuális értékét folyamatosan jelenítsük meg a második sor végén!
- Amennyiben 30 másodpercen belül eltaláljuk az értéket, akkor írjuk ki az első sorba, hogy ÖN NYERT, és a második sorközepére pedig a megfejtett számot (0-255).
- Ha nem sikerül 30 másodperc alatt megtalálni a megfejtést, akkor írjuk ki az első sorba, hogy ÖN VESZTETT, és a második sorba azt a számot, amit kerestünk.
- Nyereség vagy veszteség után a C gombbal induljon újra a játék!

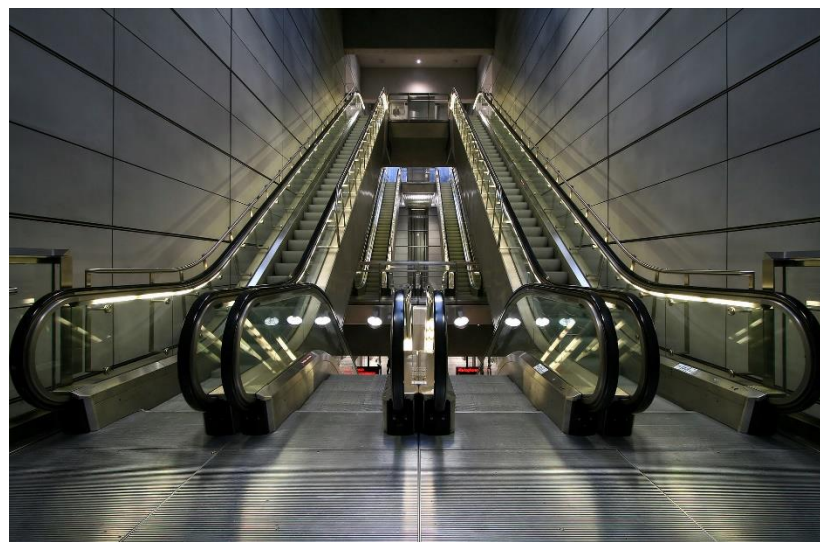


Jelfeldolgozás a közlekedésben, CZH - 2017. május 10. 10:15 – 12:00 ST121

NÉV: NEPTUN KÓD:.....

Készítsen C programot, amely a potenciométer segítségével egy mozgólépcső működését szimulálja és jelzi az üzemállapotokat a virtuális kijelzőn a következőképpen:

- A lépcső bekapcsolás után álló helyzetből indul, ekkor a LÉPCSŐ ÁLL felirat olvasható az első sorban!
- A NumPad A gombjának megnyomására a mozgólépcső elindul. Ha a P4.1 ON, akkor ELORE, ha OFF, akkor hátramenetben közlekedik, ahol az irány az első sor végén kerül kiírásra! Irányváltásra csak álló helyzetben van lehetőség. Ha a sebesség 0, akkor csak ismét az A gomb lenyomására indul el a lépcső!
- A mozgólépcső sebességét a potenciométer tekerésével lehet szabályozni 0-19.800 km/óra között. A sebesség 3 tizedesjegy pontossággal kerül kijelzésre a második sor végén a következő formában: v= yy.xxx km/ora. A sebességet a program 0,2 másodpercenként számítja. Az időzítést a Timer 0 valósítja meg.
- A B, C vagy D gombok megnyomását követően 0,8 másodpercre a lépcső vészmegállást hajt végre, azaz a sebessége leesik 0 km/óra-ra, és az első sorban a VESZMEGALLAS felirat látható! Vészmegállás után, ha a potméter ismét alaphelyzetben van, akkor az A gomb megnyomásával a lépcső ismét elindul!

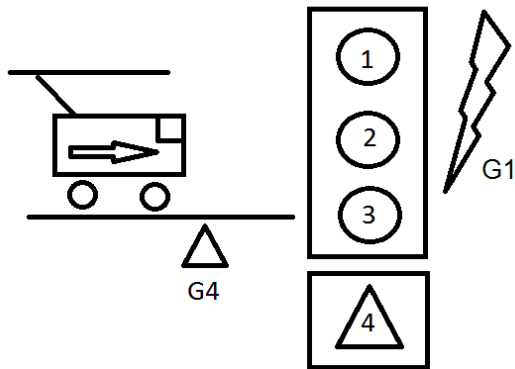


NÉV: NEPTUN KÓD:.....

Készítsen C programot, amely egy jelzőlámpát modellez, a következők figyelembe vételével. A lámpa alaphelyzetben piros fényt ad (LED1), a hurokdetektoron történő bejelentkezés (G4-es gomb lenyomása) után 1,5 másodperccel ad piros – sárgát (LED1 és LED2), majd újabb 3 másodperc múlva ad zöldet (LED3). A zöld jelzés 9 másodpercig él, majd 3 másodperces sárga idő következik, végül visszatér alapállapotba. A bejelentkezés alatt a LED4-es fény világít (észrevettük, hogy van jármű a detektoron). G1 hatására sárga villogó üzemmód áll be, 1,5 másodperces villogási idővel. Újabb G1 hatására a jelzőlámpa visszaáll alapállapotba (piros fény világít). Az órajel legyen 2 MHz, az időzítéshez használjon Timer-t!

A kijelzőn az alábbi feliratok kerüljenek kijelzésre:

- Az első sor elején az égő fények színe: ZOLD, SARGA, PIROS, PIROS-SARGA, VILLOGO SARGA.
- A második sor elején bejelentkezés után, a bejelentkezés időtartama alatt (1,5 sec): BEJELENTEKZVE.
- Az első sor végén a normál üzemmód teljes ciklusából (16,5 sec) hátralévő idő, 0,5 másodpercenként változtatva, tehát 16,5 majd 16, majd 15,5.... 0,5. Amikor piros van, akkor az idő 0, és bejelentkezés hatására indul el a visszaszámlálás.



NÉV: NEPTUN KÓD:.....

A/Készítsen C programot, amely a Mikulás december 6-i körútját szimulálja! A Téraló (# - 0x23) a kijelző felső sorában a potenciométer segítségével folyamatosan tud mozogni. A feladat az, hogy a Mikulás a KeyPad A, B, C vagy D gombjának hatására egy ajándékot helyezzen el a második sorban, mindig arra a helyre, ami felett éppen áll. Az ajándékot a \$ (0x24) jelzi. A Mikulásnak a feladat végrehajtására mindössze 20 másodperc áll rendelkezésre. A játék – bekapcsolás után a KeyPad * jelű gombjának megnyomására indul.

A kijelzőn az alábbi feliratok kerüljenek kijelzésre:

- Bekapcsoláskor: „Nyomja meg a * gombot!”
- A program futása alatt fel kell tölteni az alsó sort \$ karakterekkel. Az alsó sor egy karakter helyére több ajándékot is lerakhat, de ezek már nem kerülnek kijelzésre!
- Ha sikerül feltölteni a második sort 20 másodperc alatt, akkor az utolsó ajándék lerakása után a program leáll, és az első sorban a „Gratulálunk! üzenet kerül kijelzésre. Egy gombnyomásra csak 1 ajándékot lehet letenni!
- Ha nem sikerül feltölteni a második sort 20 másodperc alatt, akkor az időzítés lejártá után az első sorba „Virgácsot kap! kerül kijelzésre.
- Amíg egy játék fut, a program nem indítható újra a * gombbal. Ha egy játék lefutott (akár sikeresen, akár nem) akkor a játék a reset hatására indul újra!
- A kiírandó karaktertömböket **const char változonev[]=" "**; -ként definiáljuk!!!! (pl: const char nyomt[]="nyomtatas";)



Mechatronika és mikroszámítógépek, CZH - 2017. december 6. 12:15 – 14:00 ST122

NÉV: **NEPTUN KÓD:**.....

B/Készítsen C programot, amely során a Mikulás zsákját kell feltölteni. A zsák mérete 500 tonna. A zsákot a numerikus billentyűzet segítségével lehet feltölteni, ahol gombnyomásra annyi tonnával növekedik a zsák tartalma, amekkora a gombhoz tartozó érték (csak az 1-9 közötti gombnyomás érvényes). A cél az, hogy a zsákot teletöltsük. A töltésre a potenciométer segítségével beállított időzítés áll rendelkezésre, amely 15-30 másodperc között változhat (A potenciométer bal állásában 15, a jobb állásában 30 másodperc áll rendelkezésre).

A program a következőképpen működik:

- Bekapcsoláskor: „Nyomja meg a * gombot!” felirat látható. A gombnyomás hatására indul el a játék.
- Indulás után elsőként beolvassuk a potenciométerrel beállított értéket, és kiírjuk két számjegyes formátumban a második sor elejére (xx másodperc). A másodperc érték folyamatosan csökken, kvázi mint egy visszazámláló óra!
- A futás során az első sorba írjuk ki a zsák aktuális állapotát (YYY tonna), és gondoskodni kell róla, hogy egy gombnyomást a program csak 1* számoljon. A zsákban nem lehet több ajándék, mint 500 tonna.
- Az időzítés lejártá után megnézzük, hogy sikerült-e a zsákot megtölteni. Ha több van benne, mint 496 tonna, akkor írjuk ki, hogy „Nyert!”, ha kevesebb, akkor „Nem nyert!”
- Egy futás után a program csak a reset gombbal indítható újra!
- A kiírandó karaktertömböket const char valtozonev[]=” ”; -ként definiáljuk!!!! (pl: const char nyomt[]=”nyomtatas”);



Mechatronika és mikroszámítógépek, CPZH - 2017. december 12. 16:00 – 17:30 ST122

NÉV: **NEPTUN KÓD:**.....

PZH/Készítsen C programot, amely egy autó sebességváltóját (üresből 5-ösbe) szimulálja a potenciométer és a NumPad segítségével. Az autó sebessége 0-130 km/h között szabályozható, a potenciométer alsó állásában 0, a felső állásában 130 km/h, ahol ez az érték folyamatosan kijelzésre kerül a második sorban. A sebességfokozatok az 1-5 nyomógombok segítségével, a következőképpen valósul meg:

A kijelzőn az alábbi feliratok kerüljenek kijelzésre:

- Bekapcsoláskor: „Nyomja meg a * gombot!” kijelzés látható a kijelzőn. A * gomb megnyomására a jármű bekapcsol, és üresbe kerül, a kijelző első sorsában „ures” felirat látható, ha a sebesség kisebb, mint 1 km/h. Ha nagyobb, a jármű nem tehető üresbe, várjuk az újbóli gombnyomást.
- Sebességváltásra az alábbi értékek [km/h] vonatkoznak:

| | | | | | |
|--------------------|---|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 |
| felfelé (legalább) | 0 | 10 | 30 | 50 | 60 |

- **Csak felfelé lehet váltani a programban!**
- A váltás úgy valósul meg, hogy a kívánt fokozat gombját min. 2 másodpercig nyomva tartjuk. Ha a gombnyomás ideje rövidebb, mint 2 másodperc, a kocsifullad, és ez kerül kiírásra. Lefulladás után a program a reset hatására indulhat el újra. (A lenyomásvizsgálathoz használjon timer-t!)
- Egyszerre csak 1 fokozatot lehet ugrani, nincs lehetőség pl. 1-esből 3-as fokozatba váltásra.
- A kiírandó karaktertömböket const char valtozonev[]=” ”; -ként definiáljuk!!!! (pl: const char nyomt[]=”nyomtatas”);



Jelfeldolgozás a közlekedésben, CZH - 2018. május 16. 10:15 – 12:00 ST122

NÉV: NEPTUN KÓD:.....

Készítsen C programot, amely egy rotációs telefon működését szimulálja a potenciométer és a numpad segítségével. A potenciométer összesen 11 értéket tud mutatni, ebből a bal szélső helyzet jelenti az A (alaphelyzetet), majd következik 10 db számjegy (0-9), ahol a 9-es tartozik a jobb szélső állapothoz. A program csak akkor kapcsol be a reset lenyomása után, ha a potméter a bal szélső állásban van. A potméter állása folyamatosan az első sor első helyére kerül kiírásra.

A program a következőképpen működik:

- Bekapcsolás után a numpad # gombjának lenyomására elindul a tárcsázás. A tárcsázásra összesen 20 másodperc áll rendelkezésre.
- A potméter tetszőleges állásába elforgatva tudjuk a számjegyeket betárcsázni úgy, hogy a kívánt pozícióban az A jelű gombot megnyomva rögzül a számjegy a második sorba. Ha egy számjegyet rögzítettünk, akkor vissza kell tekerni a potmétert a bal szélső állásba (A-alaphelyzet), majd jöhet a következő számjegy. (Az A tárcsázása nem lehetséges).
- Egy hívószám 4 db tetszőleges számjegyből áll. Ha a 4. számjegyet is betárcsáztuk a rendelkezésre álló idő alatt, akkor az utolsó számjegy betárolása után 6 másodpercig a „HIVAS” üzenet, majd ez után a „HANGPOSTA” üzenet jelenik meg az első sor végén. Ebből az állapotból csak reset hatására tud a program kilépni, és minden kezdődik elölről.
- Ha 20 másodperc alatt nem sikerül betárcsázni a 4 számjegyet, akkor a „HIBA” üzenet jelenik meg a második sorban a számjegyek helyén, a program ebből az állapotból szintén csak reset hatására lép ki.



Jelfeldolgozás a közlekedésben, CPZH - 2018. május 22. 14:15 – 16:00 ST122

NÉV: NEPTUN KÓD:.....

PZH Készítsen C programot, amely a potenciométer segítségével egy játékot szimulál a következőképpen:

- A program elején írjuk ki a kijelzőre, hogy: Kezdshez nyomja meg a C gombot!
- A C gomb lenyomása után a program beolvassa és eltárolja a potenciométerrel beállított értéket (0-255 közötti szám). A beolvasás után a felhasználónak 30 másodperce van kitalálni, a P4-es kapcsolók megfelelő beállításával a potméterről beolvasott értéket.
- A P4-es porton a P4.0 a 2^0 , a P4.1 a 2^1 , a P4.2 a 2^2 (stb) értéknek felel meg, ha a kapcsoló állapota ON, azaz az a cél, hogy a kapcsolók megfelelő beállításával megtaláljuk az aktuálisan a potméterről beolvasott értéket.
- A kijelzőn folyamatosan írjuk ki, hogy a P4-es porton beállított kombináció a keresett számnál kisebb vagy nagyobb, és a P4-es port aktuális értékét folyamatosan jelenítsük meg a második sor végén!
- Amennyiben 30 másodpercen belül eltaláljuk az értéket, akkor írjuk ki az első sorba, hogy ÖN NYERT, és a második sorközepére pedig a megfejtett számot (0-255).
- Ha nem sikerül 30 másodperc alatt megtalálni a megfejtést, akkor írjuk ki az első sorba, hogy ÖN VESZTETT, és a második sorba azt a számot, amit kerestünk.
- Nyereség vagy veszteség után a C gombbal induljon újra a játék!



NÉV: NEPTUN KÓD:.....

ZH/A Készítsen C programot, amely egy vízforraló működését modellezi a következőképpen:

- A berendezést alaphelyeztben kapcsoljuk be, ahol a kijelző első sorában a WAITING felirat látható.
- A vízforralóban lévő víz mennyisége 0,15-1,5 liter között változhat, amit a potenciométerrel lehet beállítani. A mennyiség kijelzése két tizedesjegy pontossággal történjen a második sor elején.
- A P5.2-es nyomógomb megnyomására elindul a melegítés, ahol a melegítéshez szükséges idő $T [\text{sec}] = \text{a forralóban lévő vízmennyiség} [l] * 20$. A melegítés alatt a 2. sor végén a hátralévő időt írjuk ki másodpercben, 1 tizedesjegy pontossággal. Az első sorban ez alatt az IN PROGRESS felirat látható. Az időzítés lejártá után a víz felforr, és a READY felirat látható.
- A forralás alatt a vízmennyiség nem változhat, ha mégis, akkor a folyamat leáll, és ERROR felirat kerül kijelzésre.
- A melegítés után a P5.3-as nyomógomb megnyomására térhetünk át alapállapotba.
- A kiírandó karaktertömböket const char változonev[]=" "; -ként definiáljuk!!!! (pl: const char nyomt[]="nyomtatas");



NÉV: NEPTUN KÓD:.....

ZH/B Készítsen C programot, amellyel egy labirintusos játékot lehet játszani a következőképpen:

- A játékot alaphelyeztben kapcsoljuk be, ekkor a 'Kezdéshez nyomja meg a * gombot!' kijelzés látható.
- A gomb megnyomása után egy labirintus rajzolódik ki, ahol a '|' karakter (0x7C) jelöli a falakat. A falak helye fix. A bal felső sarokban a játékost jelölő 'O' látható, míg a jobb alsó sarokban a célt jelölő 'C' látható.

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|
| O | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | C |
| 0. | 1. | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | 10. | 11. | 12. | 13. | 14. | 15. |

- A NumPad 2 (fel), 4 (balra), 5 (le), és 6 (jobbra) nyomógombjaival lehet mozogni a pályán, a falak helyére nem lehet lépni.
- A játék elindítása után (* megnyomása) el kell érni a célt úgy, hogy 2 lépés között nem telhet el, csak maximum 5 másodperc.
- Ha elértük a célt, akkor az 'Ön nyert' felirat látható. Ha lejár a két gombnyomás közötti 5 másodperc, akkor az 'Ön veszett' felirat látható.
- A játék a reset megnyomására indulhat újra!
- A kiírandó karaktertömböket const char változonev[]=" "; -ként definiáljuk!!!! (pl: const char nyomt[]="nyomtatas");



NÉV: NEPTUN KÓD:.....

ZH/C Készítsen C programot, amely egy kávéfőző működését modellezi a következőképpen:

- A berendezés 5 különböző kávékészítményt tud lefőzni, melyek: ristretto, espresso, capuccino, latte macchiato és koffeinmentes.
- Bekapcsoláskor az első sorba kerüljön kiírásra az aktuálisan választható típus. A típusok között a potenciométer tekerésével lehet választani.
- A numerikus billentyűzet 'A' gombjának lenyomására elindul a főzés folyamata. Elsőként a 2. sorba kiírásra kerül a következő felhívás: CUKOR MENNYISÉGE. Ekkor a 'B' (cukor nélkül), 'C' (egy cukorral), vagy 'D' (két cukorral) gombok megnyomására kiírásra kerül a 2. sor elején a beállított mennyiség. A 'B', 'C' vagy 'D' gombok valamelyikének megnyomására 5 másodperc áll rendelkezésre. Ha az idő letelik, alapállapotba térünk vissza, újra lehet kávétypust választani.
- A cukor kiválasztása után elindul a főzési folyamat, ami 8 másodpercig tart. Ezalatt az idő alatt a 2. sor végén egy W látható. A főzés lejártát villogó LED-ek (P5.4-P5.7) jelzik 1,5 másodperces villogással, és a 2. sor végén egy R betű látható.
- A program alapállapotba a reset lenyomására lép át.
- A kiírandó karaktertömböket const char valtozonev[]=" "; -ként definiáljuk!!!! (pl: const char nyomt[]="nyomtatas");



NÉV: NEPTUN KÓD:.....

ZH/D Készítsen C programot, amellyel egy autós játékot lehet játszani a következőképpen:

- A játékot alaphelyeztben kapcsoljuk be, ekkor a 'Kezdéshez nyomja meg a * gombot!' kijelzés látható.
- A gomb megnyomása után egy versenypálya rajzolódik ki, ahol a felső sor az ellenfélé, az alsó az Öné. Mindkét játékost egy '>' karakter (0x3E) jelez, és mindketten a bal szélről indulnak.

| | | | | | | | | | | | | | | | | | | | |
|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| > | | | | | | | | | | | | | | | | | | | |
| > | | | | | | | | | | | | | | | | | | | |

- A képernyő megjelenése után indul a játék. Az ellenfél 1 0,9 másodperc/kocka sebességgel mozog jobbra, míg a játékos autója 0,45 – 1,35 másodperc/kocka sebességgel mozog jobbra, ahol a sebességet a potenciométer forgatásával lehet beállítani egyenlő felbontással (bal szélső állás 0,45 másodperc, jobb szélső állás 1,35 másodperc). A sebesség folyamatosan változtatható, az aktuális sebességértéket mindig az előző ugrás végén kell beállítani (induláskor úgy tekintjük, mintha történ volna egy ugrás).
- A melyik jármű előbb eléri a képernyő jobb szélét, az nyer. Ekkor az 'Ön veszett' vagy 'Ön nyert' felirat látható.
- A játék vége után a reset gombbal lehet visszatérni alapállapotba!
- A kiírandó karaktertömböket const char valtozonev[]=" "; -ként definiáljuk!!!! (pl: const char nyomt[]="nyomtatas");

