| | Ciklusok 1 - if, while |
|----------|--|
| | vezérlési szerkezet, adott utasítások ismétlését hivatott kezelni for (kezdőérték; feltétel; érték léptetése) { ciklusmag |
| | pl.: for (i=1; i<11; i++) |
| | { utasitások } For ciklus |
| | i = 0 $i = 11$ |
| In [16]: | <pre>#include <stdio.h> int main(){ for (int i = 0; i < 11; i++){ printf("%d ", i); }</stdio.h></pre> |
| | <pre>printf("\nKész!"); } 0 1 2 3 4 5 6 7 8 9 10 Kész!</pre> |
| | <pre>While ciklus i = 0 i = 11</pre> |
| In [17]: | <pre>#include <stdio.h> int main(){ int x = 0; while (x < 11){</stdio.h></pre> |
| | <pre>printf("%d ", x);</pre> |
| | Kész! Do While ciklus |
| In [19]: | <pre>i = 0 i = 11 #include <stdio.h></stdio.h></pre> |
| | <pre>int main(){ int i = 0; do { printf("%d ", i); i++; } }</pre> |
| | <pre>while (i < 11); printf("\nKész!"); } 0 1 2 3 4 5 6 7 8 9 10 Kész!</pre> |
| | Példa - Egymásba ágyazott ciklusok Írassa ki a 10*10-es szorzótábla eredményeit! |
| | formázza be a megadott minta alapján jelenítse meg a műveleteket is Minta: |
| | 1 2 3 4 5 6 7 8 9 10 1 1 2 3 4 5 6 7 8 9 10 2 2 4 6 8 10 12 14 16 18 20 3 3 6 9 12 15 18 21 24 27 30 4 4 8 12 16 20 24 28 32 36 40 |
| | 5 5 10 15 20 25 30 35 40 45 50 6 6 12 18 24 30 36 42 48 54 60 7 7 14 21 28 35 42 49 56 63 70 8 8 16 24 32 40 48 56 64 72 80 |
| | 9 9 18 27 36 45 54 63 72 81 90 10 10 20 30 40 50 60 70 80 90 100 Minta: |
| | 1 2 3 4 5 6 7 8 9 10 1 1 * 1 = 1 1 * 2 = 2 1 * 3 = 3 1 * 4 = 4 1 * 5 = 5 1 * 6 = 6 1 * 7 = 7 1 * 8 = 8 1 * 9 = 9 1 * 10 = 10 2 2 * 1 = 2 2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = |
| | 18 2 *10 = 20 3 3 * 1 = 3 3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 = 27 3 *10 = 30 4 4 * 1 = 4 4 * 2 = 8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36 4 *10 = 40 |
| In [2]: | <pre>#include <stdio.h> int main(){ printf("%3c ", ' ');</stdio.h></pre> |
| | <pre>for (int i = 1; i <= 10; i++) { printf("%3d ", i); } printf("\n%3c ", ' '); for (int i = 1; i <= 10; i++) { printf("%3c ", '-');</pre> |
| | <pre> } for (int i = 1; i <= 10; i++){ printf("\n%3d ", i); for (int j = 1; j <= 10; j++){ printf("%3d ", i*j); } } </pre> |
| | } 1 2 3 4 5 6 7 8 9 10 |
| | 2 2 4 0 8 10 12 14 10 18 20 3 3 6 9 12 15 18 21 24 27 30 4 4 8 12 16 20 24 28 32 36 40 5 5 10 15 20 25 30 35 40 45 50 6 6 12 18 24 30 36 42 48 54 60 7 7 14 21 28 35 42 49 56 63 70 8 8 16 24 32 40 48 56 64 72 80 |
| | 9 9 18 27 36 45 54 63 72 81 90 10 10 20 30 40 50 60 70 80 90 100 1. Feladat |
| | Készítsen programot ciklus segítségével, ami kiírja az első 10 számot! Módosítsa úgy a programot, hogy a felhasználó határozza meg a kiírt számsor hosszát A program csak a páratlan számokat írja ki a ciklus segítségével |
| In []: | <pre>#include <stdio.h> int main (void) { int i;</stdio.h></pre> |
| | <pre>printf("Ez a program ciklus segítségével kiírja az első 10 számot.") for (i=1; i<11; i++) { printf("%d ", i); }</pre> |
| | return 0; } |
| | 2. Feladat |
| | Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! Bővítse a programot, hogy a felhasználó döntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) |
| In []: | Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! Bővítse a programot, hogy a felhasználó döntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) Az eredmény kiíratása után ajánlja fel a felhásználó számára az újbóli futtatás lehetőségét A program legyen képes kiírni az elvégzett műveletet is #include <stdio.h></stdio.h> int main (void) |
| In []: | Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! Bővítse a programot, hogy a felhasználó döntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) Az eredmény kiíratása után ajánlja fel a felhásználó számára az újbóli futtatás lehetőségét A program legyen képes kiírni az elvégzett műveletet is #include <stdio.h> int szam, i; int faktorialis = 1; int szum = 0; int state; </stdio.h> |
| In []: | Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! Bővítse a programot, hogy a felhasználó döntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) Az eredmény kiíratása után ajánlja fel a felhásználó számára az újbóli futtatás lehetőségét A program legyen képes kiírni az elvégzett műveletet is #include <stdio.h> int szam, i; int faktorialis = 1; int szum = 0; int state; printf("A program visszaadja a határértékig előforduló egész számok összegét és a bevitt szám faktoriálisát printf("Adj meg egy pozitív egész számot: "); scanf("%d", &i); szam=i; printf("0 - faktoriális\n1 - összeg"); </stdio.h> |
| In []: | Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! Bővítse a programot, hogy a felhasználó döntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) Az eredmény kiíratása után ajánlja fel a felhásználó számára az újbóli futtatás lehetőségét A program legyen képes kiírni az elvégzett műveletet is #include <stdio.h> int main (void) { int szam, i; int faktorialis = 1; int szum = 0; int state; printf("A program visszaadja a határértékig előforduló egész számok összegét és a bevitt szám faktoriálisát printf("Adj meg egy pozitív egész számot: "); scanf("%d", &i); szam=i;</stdio.h> |
| In []: | <pre>Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! • Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! • Bővítse a programot, hogy a felhasználó döntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) • Az eredmény kiíratása után ajánlja fel a felhásználó számára az újbóli futtatás lehetőségét • A program legyen képes kiírni az elvégzett műveletet is #include <stdio.h> int main (void) { int szam, i; int faktorialis = 1; int szum = 0; int state; printf("A program visszaadja a határértékig előforduló egész számok összegét és a bevitt szám faktoriálisát printf("Adj meg egy pozitív egész számot: "); scanf("%d", &i); szam=i; printf("0 - faktoriális\n1 - összeg"); scanf("%d \n", &state); if (state==0) { for(i; i >= 1; i) { faktorialis *= i; } }</stdio.h></pre> |
| In []: | <pre>Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! • Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! • Bővítse a programot, hogy a felhasználó döntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) • Az eredmény kiíratása után ajánlja fel a felhásználó számára az újbóli futtatás lehetőségét • A program legyen képes kiírni az elvégzett műveletet is #include <stdio.h> int main (void) { int szam, i; int faktorialis = 1; int szum = 0; int state; printf("A program visszaadja a határértékig előforduló egész számok összegét és a bevitt szám faktoriálisát printf("Adj meg egy pozitív egész számot: "); szam=i; printf("0 - faktoriális\nl - összeg"); scanf("%d', %i); szam=i; if (state==0) { for(i; i >= 1; i) { faktorialis *= i; } printf("%d! = %d", szam, faktorialis); } else { for(i; i >= 1; i) { } } } }</stdio.h></pre> |
| In []: | Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! • Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! • Bővítse a programot, hogy a felhasználó döntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) • Az eredmény kiíratása után ajánlja fel a felhásználó számára az újbóli futtatás lehetőségét • A program legyen képes kiírni az elvégzett műveletet is #include «stdio.h» int main (void) { int szam, i; int faktorialis = 1; int szam, a; int faktorialis = 1; int szam = 0; int state; printf("A program visszaadja a határértékig előforduló egész számok összegét és a bevitt szám faktoriálisát printf("Ad) meg egy pozitiv egész számot: "); szam=1; printf("0 - faktoriális\n1 - összeg"); szam("\dd \n", &state); if (state==0) { for(i; i >= 1; i) { faktorialis *= i; } printf("\dd! = \dd", szam, faktorialis); } else { for(i; i >= 1; i) { szum += i; } printf("\dd! = \dd", számok összege \dd-ig: \dd \n", szam, szum); } return 0; } 3. Feladat Készítsen programot, aminek ciklusa 1-től 100-ig megy és megvizsgálja a számokat hárommal és öttel való oszthatóság |
| In []: | Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! • Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! • Bővítse a programot, hogy a felhasználó dőntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) • Az eredmény kiiratása után ajánlja fel a felhásználó számára az újbóli futtatás lehetőségét • A program legyen képes kiírni az elvégzett műveletet is #intlude <stódio.h> int main (void) { int szam, i; int faktorialis = 1; int szam = 0; int state; printf("A program visszaadja a határértékig előforduló egész számok összegét és a bevitt szám faktoriálisát szamí("adj meg egy pozitív egész számot: "); szamí("adj meg egy egy pozitív egész számot: "); szamí("adj meg egy egy egy egy egy egy egy egy egy e</stódio.h> |
| | Készítsen programot, ami a felhasználótói bekér egy számot, majd visszaadja az addig található számok összegét! Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! Bóvitse a programot, hogy a felhasználó döntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) Az eredmény kiiratása után ajánija fel a felhásználó számára az újbóli futtatás lehetőségét A program legyen képes kiirni az elvégzett műveletet is #include vstdio.h> int main (void) { int szam, i; int raktorialis = 1; int szam = 0; int state; printf("A program visszaadja a határértékig előforduló egész számok összegét és a bevitt szám faktoriálisát printf("Ad) meg egy pozitív egész számot: "); szamí("vd", 61); printf("B - faktoriális\n1 - összeg"); szamí("vd", 63); ja faktoriális "a i; } printf("\n4 : a i; i · ·) { faktorialis "a i; } printf("\n4 : a i; i · ·) { fattorialis "a i; } printf("\n4 : a i; i · ·) { for(i; i = 1; i · ·) { fun(i; i = 1; i · ·) { for(i; i = 1; |
| | Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadíja az addig található számok összegét! Egészítse ki a programot úgy, hogy a bevitt szám faktoriálisát kiszámolja! Bővitse a programot, hogy a felhasználó döntse el, milyen műveletet szeretne elvégezni (összeg, vagy faktoriális) A zeredmény kiiratása után ajánlja fel a felhásználó számára az újbóli futtatás lehetőségét A program legyen képes kiími az elvégzett műveletet is #include «stdio.h» int main (vold) int szam, i; int faktorialis = 1; int szam, a; int faktorialis = 1; int szam e 0; int state; printf("Adj meg egy pozitív egész számot: "); szam("Ad", 5.1); szam("Ad", 5.1); szam("Ad \n", 6.1); szam("Ad |
| | Kesztisen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! • Egésztise ki a programot, hogy a lehlasználót dintse el, milyem műveletet szeretne elvégezni (összeg, vagy laktoriális) • Az eredmény kiratása után ajánija fel a felhásználót számára az újbóli futtatás lehetőségét • A program legyen képes kirin laz elvégzett műveletet is #int talátoriala = 1; int szam, :) int faktorialis = 1; int szam, :) int szam = 0; int szam |
| | Készítsen programot, ami a felhasználótól bekér egy számot, majd visszaadja az addig található számok összegét! • Egészítse ki a programot úgy, hogy a bevit szám faktoriálisák kiszámója! • Bővitse a programot, hogy a felhasználó dőntse el, nilyen műveletet szeretne elvégezni (összeg, vagy faktoriális) • Az rezdemény kiriztása utal a ajánja fel a felhasználó számára az újbóli futtatás lehetőségét • A program legyen képes kifini az elvégzett műveletet is szinctude «stdio.h» int salm (veid) { int szam, i; int szam, i; int szam, = 0; int szam = |
| | Készítsen programot, ami a felhasználódó bekér egy számot, majd visszaadja az addig található számok összegét! - Győszíbe ki a programot, hogy a felhasználó döltse el, milyen möveletet szeretne elvégezői (összeg, vagy faktoriális) - 8 dortós a programot, hogy a felhasználó döltse el, milyen möveletet szeretne elvégezői (összeg, vagy faktoriális) - A zerodmény kirádsás údia pálnaj (el a felhásználó számára az újbóli futtatás lehetőségét - A program legyen képes kirni az elvégzett műveletet is #int toltós a millen mill |
| | Kčezilsen programot, ami a Tehnasmálóló bekér egy számot, majd visszaadja az addig találhaló számok összegét! Egészilse ki a programot hogy a felhasználó dönbe el, milyen művelétet szeretne elvégezni (összeg, vagy faktoriális) Az erdemény kirádsas utan ágánja él a felhasználó számára az újbóli futatás lehetőségét A program legyen képes kirni az elvégzett műveletet is Fánctude éstdőol.ho sint main (void) int szam.; int faktoriális. 1; int faktoriális. 1; int szám. 3; int szám. 3; int szám. 6; int szám.; faktoriális. 1; int szám. 6; int szám. |
| | Kézőksen programot, ami a felhacznádód i bekér egy számot, majd viscosadja az addig található számok ősülegétt Egészíbse ki a programot (gy. hogy a bevitt szám faktoriálisát kiszámoljat Bévitse a programot (gy. hogy a bevitt szám faktoriálisát kiszámoljat Bévitse a programot hogy a felhasznádó dönse el, milyen műveletet szeretne elvégezel (összeg, vogy faktoriális) A medmény findásás utal naján jel a felhasznádó dönse el, milyen műveletet is Binchade aktelo. hogy a megy a megy elvelet műveletet is Binchade aktelo. hogy a megy elvelet műveletet is Binchade aktelo. hogy a megy elvelet e |
| In []: | Keshtsen programut, ami a Archosznakiói bekér egy számot, mejd visososágia ar addig talaható számok ösöregett Egészítse ki a programot úgy, hogy a bevitt szám foltorrállisát kiszámójai Bolvise a programot úgy hogy a felhasználó diáhase el, milyen művelette szeretne elvégezni (összeg, vagy faktoriális) A program Regna képes kilmi az elvégezet műveletet is archador a számot ár a számot ár a számot az a kilmi kuttrás lehetőséget. A program Regna képes kilmi az elvégezet műveletet is archador a számot ár a számot ár a számot ár a kilmi a kilmi a kilmi a kilmi a számot ár a kilmi a kilm |
| In []: | Resident programs. Limit a fethissmelloto bethe egy számoti, majd visszeradla az eddig talafhato számok öszergett Essecible ki a corporanci kuy, hogy a besül szám takorátiliski kiszámold. Biszkise ki a corporanci kuy, hogy a besül szám takorátiliski kiszámold. Biszkise ki a corporanci kuy a nithakoráti dátozo init promiscient számora promiscient számora orgáni kiszámold. Az ecendeny kirasása usan jadla ár in zhákárnál számára az újháli nitratás ichetőségét A program szám kellő kiszámold kiszámold számára az újháli nitratás ichetőségét A program szám szám számára |
| In []: | Acception programs, and a tehsomolifoni belier agy stimen, majd vissosodja an addig tatilizad station destroys. Buschbe sit across and control to the statilization of the stati |
| In []: | Residuen programot, amine fedinacialodo peter very scannot, maid viscosade as autori cale had scannot desceptif • Electrica programos hogy interaction factoristics, tickerinolis; • Reforman programos hogy sinteraction descent employment reserves enleggers inserved, segy transmission. • An extendinely kinetic claim signified of a fembraneai activities in the contempts are contempts. • A recondency kinetic claim signified of a fembraneai activities in the contempts are contempts. • A recondency kinetic claim in the contempts of the contempts |
| In []: | Kinastern programot, amila followarishto behör egy salmot, majd visuseadia ac edida talahata salmot bascapet - (Spettate dia programot day), majo a horita sain internalisht kinat majot October programot, florar y fillowarishto data de minjem modellose sainti sa cipida (futatas interdoseget) An errondeny directas ustan apida et a terhisoratio sainti sa cipida (futatas interdoseget) An errondeny Modellose in chidyper mituolicet is - Internal (1) - Internal (1 |
| In []: | Nacionary programmo, and a fellowardstot better gry scenou, mail viscandia as adolg to limino scenot of the companion Cay, a followard score of miles in the scientists of bodies and programmo Cay, a followards of some of miles in the scientists of the companion |
| In []: | Kactices grops and it, on a felloward both better any scenario, made discarding as adold talk made scenario. Is glader to be grown to hap a retinous had better any scenario and post talk discarding and scenario. As an advantage limit as elegated to a fermioscopic or scenario and post talk discarding and a design to a fermioscopic or scenario and post talk discarding and a design to a fermioscopic or scenario and post talk discarding and a design to a fermioscopic or scenario and post talk discarding and a design talk discarding an |
| In []: | Assertion programmer, and a instruction 6xth beaching yearlington, much discounted as a place on the local beaching and programmer to the |
| In []: | Continues congruence and a follown still of below any varies, may be increasing an editing distination would be expected and project and the continues of the progression of the project of the continues of the c |
| In []: | deuthors programmy, am is a filled conditional basin way seems or modify this weetly as early point of an empty of the condition of the condit |
| In []: | ideations programs, on or binasons of all activity abortion that is second to the activity of the second of the activity of th |
| In []: | Association programmed, and a telescope displaced complete and reconstruction of a second control consequence of the pays absociation and control and control |