

Programming In C (4331105) - Winter 2022 Solution

Milav Dabgar

March 28, 2023

પ્રશ્ન 1 [a ગુણ]

3 C ભાષાના મૂળભૂત data types તેમની range સાથે જણાવો

જવાબ

જવાબ:

Data Type	Size (bytes)	Range
char	1	-128 થી 127
int	2 or 4	-32,768 થી 32,767 (2 bytes) અથવા -2,147,483,648 થી 2,147,483,647 (4 bytes)
float	4	3.4E-38 થી 3.4E+38
double	8	1.7E-308 થી 1.7E+308

મેમરી ટ્રીક

"CIFD - Computer Is Fundamentally Digital"

પ્રશ્ન 1 [b ગુણ]

4 વેરીએબલના નામ માટેના નિયમો સમજાવો.

જવાબ

જવાબ:

નિયમ	ઉદાહરણ
અક્ષર અથવા અંડરસ્કોરથી શરૂ થવું જોઈએ	માન્ય: _count, name / અમાન્ય: 1score
અક્ષરો, અંકો, અંડરસ્કોર હોઈ શકે છે	માન્ય: user_1 / અમાન્ય: user-1
Keywords નો ઉપયોગ કરી શકાતો નથી	માન્ય: integer / અમાન્ય: int
Case sensitive છે	total અને TOTAL અલગ છે

આકૃતિ:

વેરિએબલ નામકરણ નિયમો

[A-Z, a-z, _] → [A-Z, a-z, 0-9, _]*

મેમરી ટ્રીક

"LUCK - Letters Underscore Case Keywords"

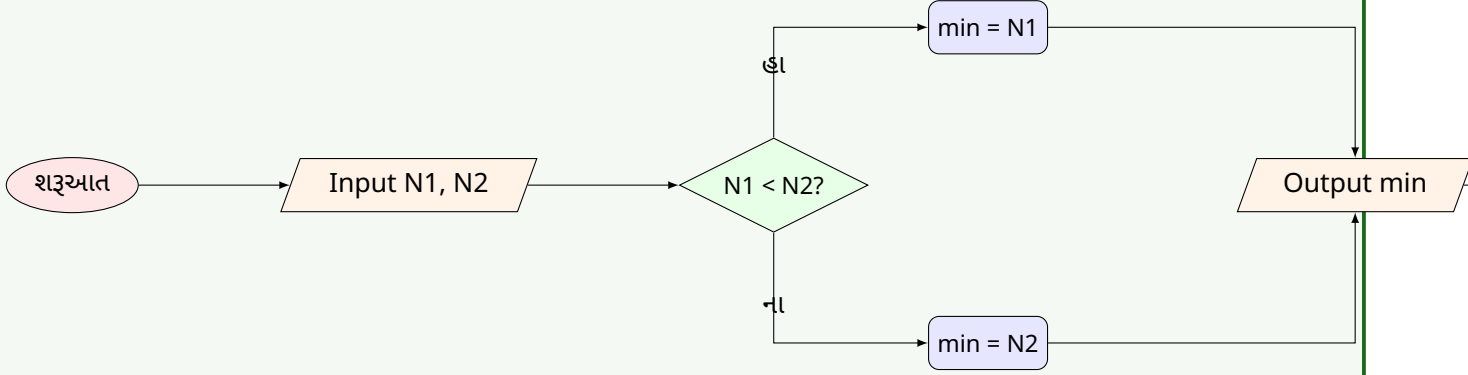
પ્રશ્ન 1 [c ગુણ]

7 ફ્લોચાર્ટની વ્યાખ્યા લખો. બે પૂર્ણાંક નંબરો N1 અને N2 માંથી લઘુત્તમ શોધવા માટેનો ફ્લોચાર્ટ દોરો.

જવાબ

જવાબ:

ફ્લોચાર્ટ એ અલ્ગોરિથમનું ચિત્રાત્મક નિરૂપણ છે જે પ્રમાણભૂત પ્રતીકોનો ઉપયોગ કરીને પ્રક્રિયાના પગલાં બતાવે છે.



- ફ્લોચાર્ટ પ્રતીકો: તાર્કિક પગલાંનું દ્રશ્ય નિરૂપણ
- નિર્ણય ડાયમંડ: પ્રવાહનો માર્ગ નક્કી કરવા શરત તપાસે છે
- પ્રક્રિયા બોક્સ: ગણતરીઓ અથવા કામગીરી સમાવે છે

મેમરી ટ્રીક

"FAST - Flow Analysis Shown Through-charts"

OR

પ્રશ્ન 1 [c ગુણ]

7 અલ્ગોરિથમની વ્યાખ્યા લખો. વર્તુળનું ક્ષેત્રફળ અને પરિઘ ગણવા માટેનો અલ્ગોરિથમ લખો.

જવાબ

જવાબ:

અલ્ગોરિથમ એ કોઈ ચોક્કસ સમસ્યા ઉકેલવા માટેની સુનિશ્ચિત સૂચનાઓની પગલાવાર પ્રક્રિયા છે.

વર્તુળની ગણતરી માટે અલ્ગોરિથમ:

1. START
2. Input radius r
3. Calculate area = $\pi * r * r$
4. Calculate circumference = $2 * \pi * r$
5. Output area, circumference
6. STOP

પગલું	કામગીરી	સૂત્ર
1	ત્રિજ્યા મેળવો	Input r
2	ક્ષેત્રફળ ગણો	$A = \pi \times r^2$
3	પરિઘ ગણો	$C = 2 \times \pi \times r$
4	પરિણામ દર્શાવો	Output A, C

મેમરી ટ્રીક

"SICS - Steps In Clear Sequence"

પ્રશ્ન 2 [a ગુણ]

3 printf() અને scanf() વચ્ચેનો તફાવત લખો.

જવાબ

જવાબ:

ફીચર	printf()	scanf()
ઉદ્દેશ	સ્ક્રીન પર ડેટા આઉટપુટ કરે છે	કીબોર્ડથી ડેટા ઇનપુટ લે છે
ફોર્મેટ	printf("format", variables)	scanf("format", &variables)
Returns	પ્રિન્ટ થયેલ અક્ષરોની સંખ્યા	સફળતાપૂર્વક વાંચેલ વસ્તુઓની સંખ્યા
એડ્રેસિંગ	વેરિએબલ નામોનો ઉપયોગ કરે છે	વેરિએબલ્સના એડ્રેસ (&var) નો ઉપયોગ કરે છે

મેમરી ટ્રીક

"IO-AR - Input Output-Address Returns"

પ્રશ્ન 2 [b ગુણ]

4 Conditional operator નો ઉપયોગ કરીને બે નંબરોમાંથી મહત્તમ શોધવા માટે C પ્રોગ્રામ લખો.

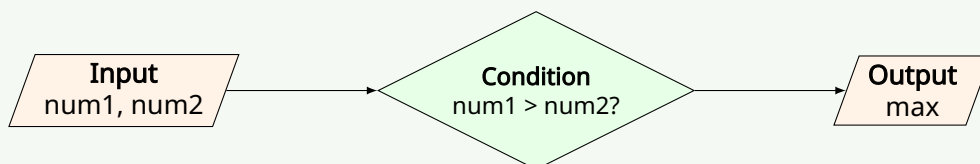
જવાબ

જવાબ:

```

1 #include <stdio.h>
2
3 int main() {
4     int num1, num2, max;
5
6     printf("Enter two numbers: ");
7     scanf("%d %d", &num1, &num2);
8
9     max = (num1 > num2) ? num1 : num2;
10
11     printf("Maximum number is: %d", max);
12
13     return 0;
14 }
```

આકૃતિ:



મેમરી ટ્રીક

"CTO - Condition Then Output"

પ્રશ્ન 2 [c ગુણ]

7 એરિથમેટિક અને રીલેશનલ ઓપરેટર્સ ઉદાહરણ સાથે સમજાવો.

જવાબ

જવાબ:

પ્રકાર	Operators	ઉદાહરણ	પરિણામ
એરિથમેટિક ઓપરેટર્સ			
સરવાળો	+	5 + 3	8
બાદબાકી	-	5 - 3	2
ગુણાકાર	*	5 * 3	15
ભાગાકાર	/	5 / 3	1 (પૂર્ણાંક ભાગાકાર)
મોડ્યુલસ	%	5 % 3	2 (શેષ)
રીલેશનલ ઓપરેટર્સ			
સમાન	==	5 == 3	0 (ખોટું)
સમાન નથી	!=	5 != 3	1 (સાચું)
મોટું	>	5 > 3	1 (સાચું)
નાનું	<	5 < 3	0 (ખોટું)
મોટું અથવા સમાન	>=	5 >= 5	1 (સાચું)
નાનું અથવા સમાન	<=	5 <= 3	0 (ખોટું)

મેમરી ટ્રીક

"ASMDCRO - Add Subtract Multiply Divide Compare Return Output"

OR

પ્રશ્ન 2 [a ગુણ]

3 ઓપરેટરની અગ્રતા (Precedence) ધ્યાનમાં રાખીને, જો expression $(25/3) * 4 - 10 \% 3 + 9/2$ નું મૂલ્યાંકન કરવામાં આવે તો દરેક સ્ટેપ અને છેલ્લો જવાબ લખો.

જવાબ

જવાબ:

પગલું	કામગીરી	ગણતરી	પરિણામ
1	કોંસ (25/3)	$25/3 = 8$ (પૂર્ણાંક ભાગાકાર)	8
2	મોડ્યુલસ $10 \% 3$	$10 \% 3 = 1$	1
3	ભાગાકાર $9/2$	$9/2 = 4$ (પૂર્ણાંક ભાગાકાર)	4
4	ગુણાકાર $8 * 4$	$8 * 4 = 32$	32
5	બાદબાકી $32 - 1$	$32 - 1 = 31$	31
6	સરવાળો $31 + 4$	$31 + 4 = 35$	35

અંતિમ જવાબ = 35

મેમરી ટ્રીક

"PEMDAS - Parentheses, Exponents, Multiplication/Division, Addition/Subtraction"

OR

પ્રશ્ન 2 [b ગુણ]

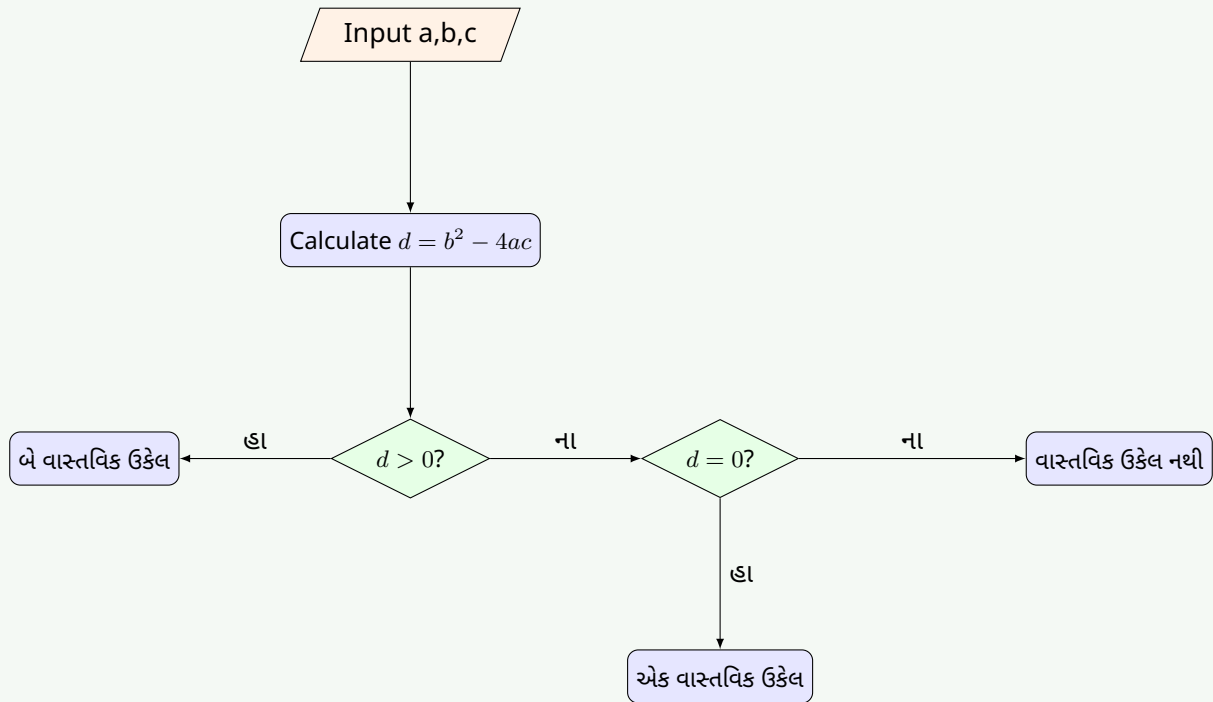
4 એલ્જેબ્રીક સમીકરણના ઉકેલ માટેનો C પ્રોગ્રામ લખો.

જવાબ

જવાબ:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     float a, b, c;
6     float discriminant, root1, root2;
7
8     printf("Enter coefficients a, b, c: ");
9     scanf("%f %f %f", &a, &b, &c);
10
11     discriminant = b*b - 4*a*c;
12
13     if (discriminant > 0) {
14         root1 = (-b + sqrt(discriminant)) / (2*a);
15         root2 = (-b - sqrt(discriminant)) / (2*a);
16         printf("Roots: %.2f and %.2f", root1, root2);
17     } else if (discriminant == 0) {
18         root1 = -b / (2*a);
19         printf("Root: %.2f", root1);
20     } else {
21         printf("No real roots");
22     }
23
24     return 0;
25 }
```

આકૃતિ:



મેમરી ટ્રીક

"QDR - Quadratic Discriminant Roots"

OR

પ્રશ્ન 2 [c ગુણ]

7 લોજિકલ અને બિટવાઈઝ ઓપરેટર્સ ઉદાહરણ સાથે સમજાવો.

જવાબ

જવાબ:

પ્રકાર	Operators	ઉદાહરણ	પરિણામ
લોજિકલ ઓપરેટર્સ			
લોજિકલ AND	&&	(5>3) && (4<7)	1 (સાચું)
લોજિકલ OR		(5<3) (4<7)	1 (સાચું)
લોજિકલ NOT	!	!(5>3)	0 (ખોટું)
બિટવાઈઝ ઓપરેટર્સ			
બિટવાઈઝ AND	&	5 & 3 (101 & 011)	1 (001)
બિટવાઈઝ OR		5 3 (101 011)	7 (111)
બિટવાઈઝ XOR	^	5 ^ 3 (101 ^ 011)	6 (110)
બિટવાઈઝ NOT	~	~5 (~ 00000101)	-6 (11111010)
Left Shift	<<	5 << 1 (101 << 1)	10 (1010)
Right Shift	>>	5 >> 1 (101 >> 1)	2 (10)

મેમરી ટ્રીક

"LAND BORNs - Logical AND OR NOT, Bitwise OR AND NOT Shift"

પ્રશ્ન 3 [a ગુણ]

3 'go to' statement નો ઉપયોગ ઉદાહરણ સાથે સમજાવો.

જવાબ

જવાબ:

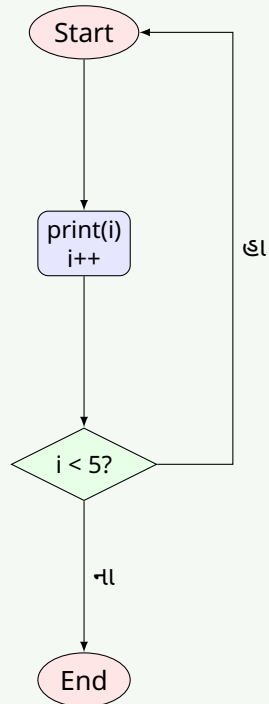
goto statement પ્રોગ્રામમાં લેબલ થયેલ statement પર સીધો જમ્પ કરવાની મંજૂરી આપે છે.

```

1 #include <stdio.h>
2
3 int main() {
4     int i = 0;
5
6     start:
7     printf("%d ", i);
8     i++;
9     if (i < 5)
10        goto start;
11
12    return 0;
13 }
14 // આઉટપુટ: 0 1 2 3 4

```

આકૃતિ:



મેમરી ટ્રીક

"JUMP - Just Unconditionally Move Program-counter"

પ્રશ્ન 3 [b ગુણ]

4 આપેલ નંબર બેકી છે કે એકી તે શોધવા માટે C પ્રોગ્રામ લખો.

જવાબ

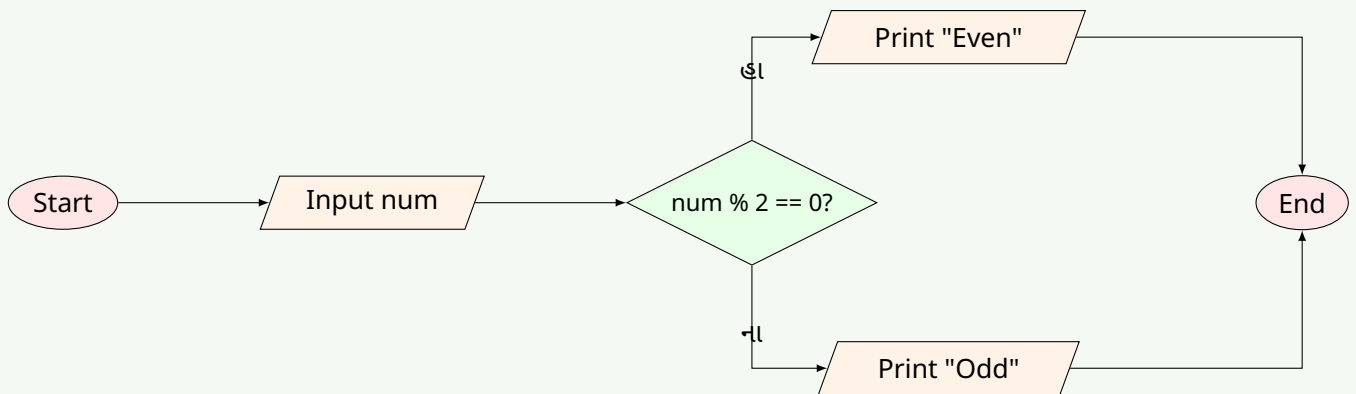
જવાબ:

```

1 #include <stdio.h>
2
3 int main() {
4     int num;
5
6     printf("Enter a number: ");
7     scanf("%d", &num);
8
9     if (num % 2 == 0)
10        printf("%d is even", num);
11    else
12        printf("%d is odd", num);
13
14    return 0;
15 }

```

આકૃતિ:



મેમરી ટ્રીક

"MODE - Modulo Odd-Even Determination"

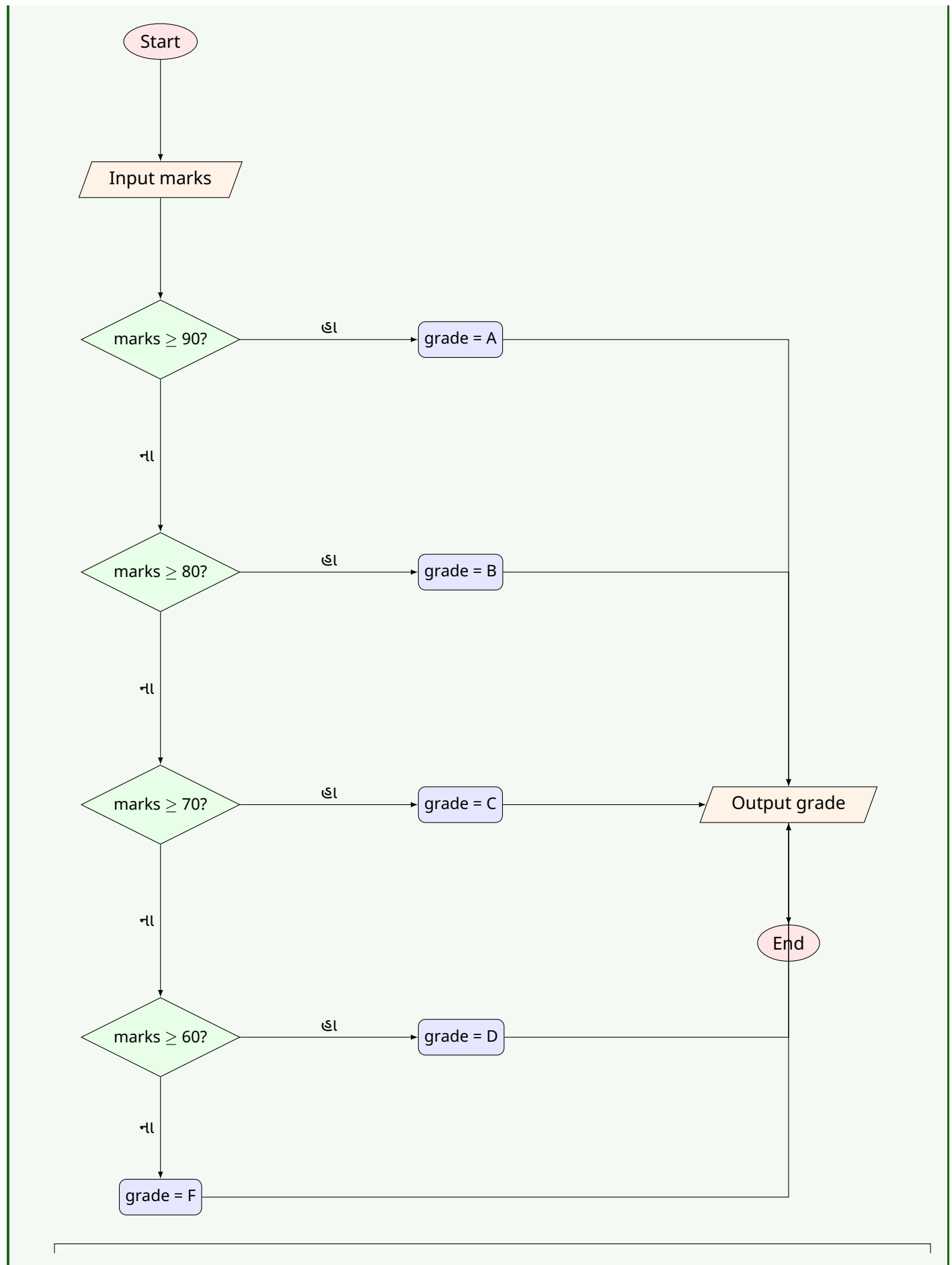
પ્રશ્ન 3 [c ગુણ]

7 else if ladder ફ્લોચાર્ટ દોરી ઉદાહરણ સાથે સમજાવો.

જવાબ

જવાબ:

Else-if ladder ક્રમમાં બહુવિધ શરતો તપાસવા માટે વપરાય છે, જેમાં પ્રથમ સાચી શરત સાથે સંકળાયેલ બ્લોક એક્ઝિક્યુટ થાય છે.



```

1 #include <stdio.h>
2
3 int main() {
4     int marks;
5     char grade;
6
7     printf("Enter marks: ");
8     scanf("%d", &marks);
9
10    if (marks >= 90)
11        grade = 'A';
12    else if (marks >= 80)
13        grade = 'B';
14    else if (marks >= 70)
15        grade = 'C';
16    else if (marks >= 60)
17        grade = 'D';
18    else
19        grade = 'F';
20
21    printf("Grade: %c", grade);
22
23    return 0;
24 }

```

- એકાધિક શરતો: ક્રમાનુસાર શરતો તપાસે છે
- પ્રથમ મેચ: માત્ર પ્રથમ સાચી શરત માટે કોડ એક્ઝિક્યુટ થાય છે
- ડિફોલ્ટ કેસ: છેલ્લે else બાકીના તમામ કેસો સંભાળે છે

મેમરી ટ્રીક

"CAFE - Condition Assess First Eligible"

OR

પ્રશ્ન 3 [a ગુણ]

3 continue અને break statement નો ઉપયોગ સમજાવો.

જવાબ

જવાબ:

Statement	હેતુ	અસર
break	લૂપ અથવા સ્વિચમાંથી બહાર નીકળવા	સમગ્ર લૂપને તરત જ સમાપ્ત કરે છે
continue	વર્તમાન પુનરાવર્તન છોડવા	લૂપના આગલા પુનરાવર્તન પર જાય છે

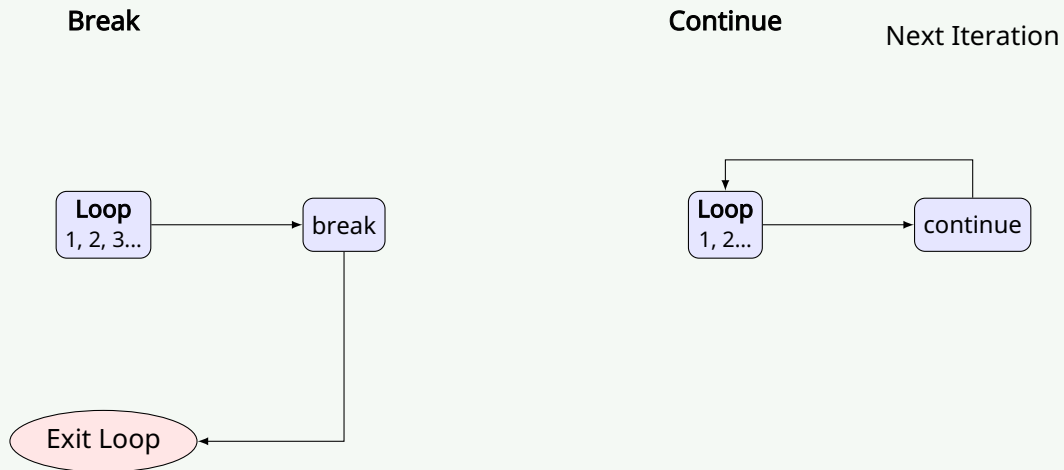
```

1 // break example
2 for(int i=1; i<=10; i++) {
3     if(i == 6)
4         break; // i=6 થાય ત્યારે લૂપમાંથી બહાર નીકળે છે
5     printf("%d ", i); // આઉટપુટ: 1 2 3 4 5
6 }
7
8 // continue example
9 for(int i=1; i<=10; i++) {
10    if(i % 2 == 0)
11        continue; // બેકી સંખ્યાઓ છોડે છે
12    printf("%d ", i); // આઉટપુટ: 1 3 5 7 9

```

13 }

આકૃતિ:



મેમરી ટ્રીક

"BEST - Break Exits, Skip with conTinue"

OR

પ્રશ્ન 3 [b ગુણ]

4 For loop નો ઉપયોગ કરીને 1 થી 10 નંબરોનો સરવાળો પ્રિન્ટ કરવા માટે C પ્રોગ્રામ લખો.

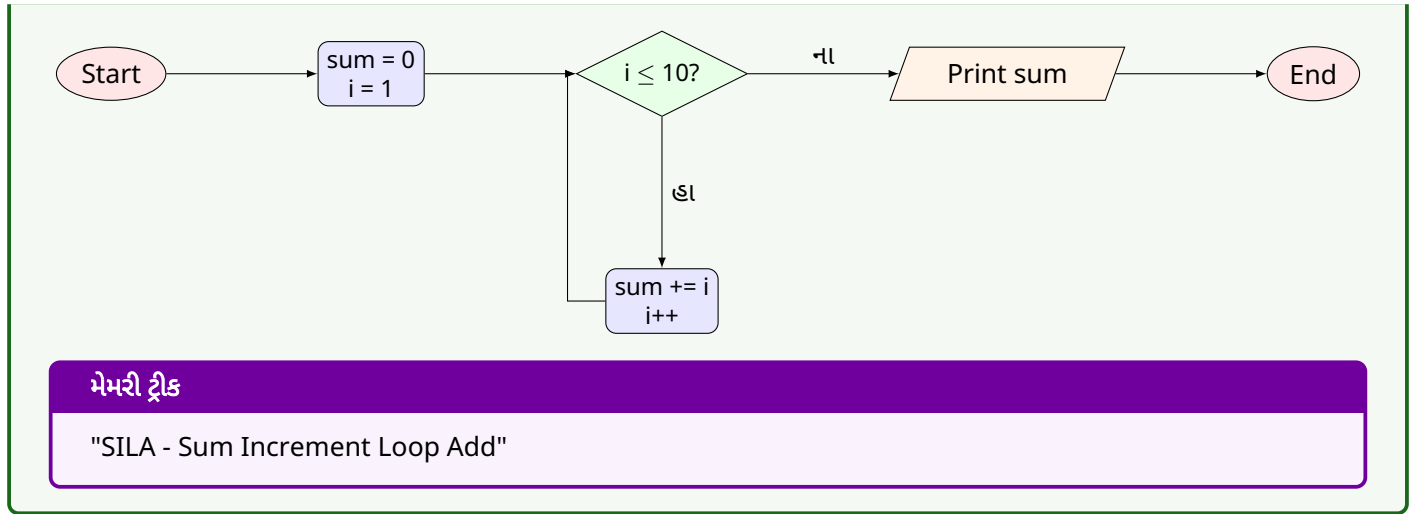
જવાબ

જવાબ:

```

1 #include <stdio.h>
2
3 int main() {
4     int i, sum = 0;
5
6     for(i = 1; i <= 10; i++) {
7         sum += i;
8     }
9
10    printf("Sum of numbers from 1 to 10: %d", sum);
11
12    return 0;
13 }
```

આકૃતિ:



OR

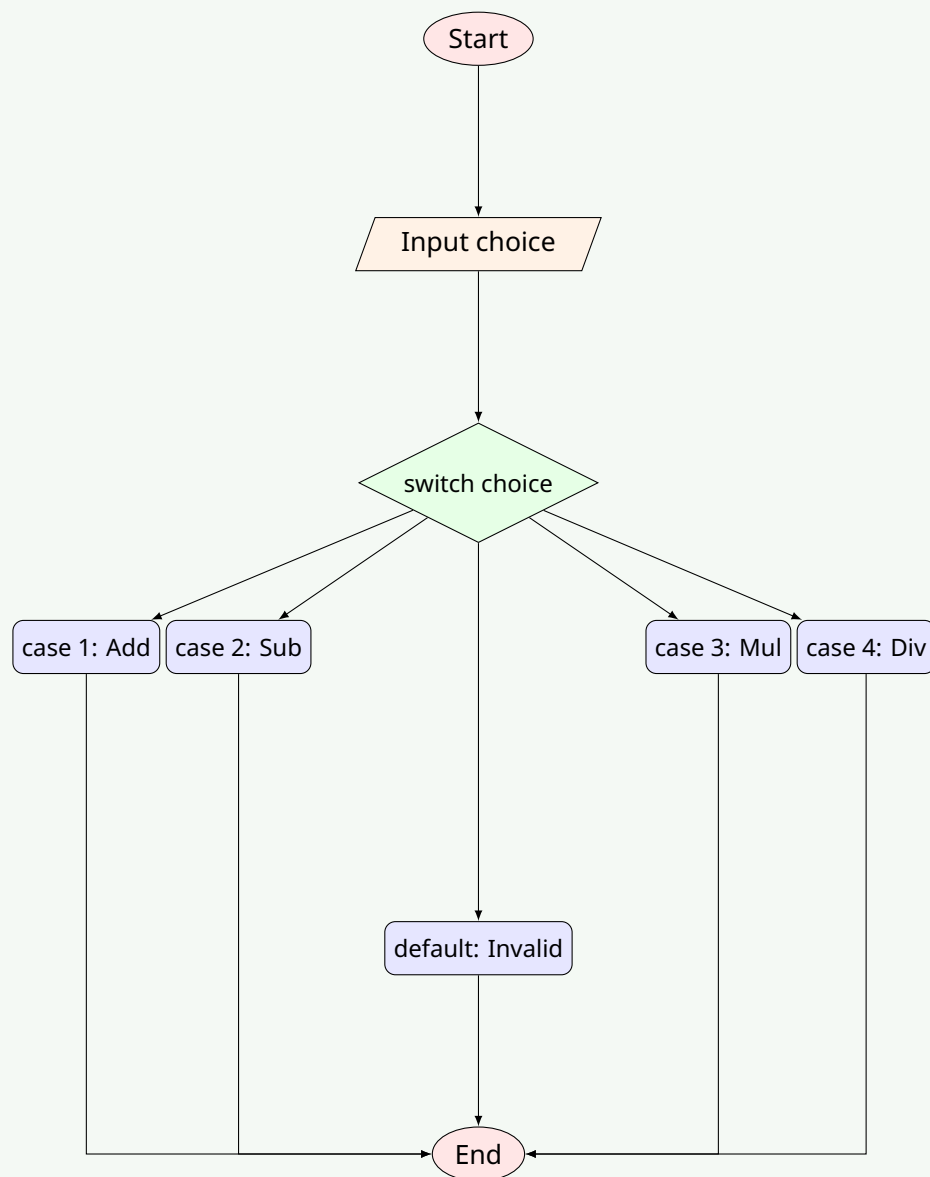
પ્રશ્ન 3 [c ગુણ]

7 સ્વિચ સ્ટેટમેન્ટ ફ્લોચાર્ટ દોરી ઉદાહરણ સાથે સમજાવો.

જવાબ

જવાબ:

સ્વિચ સ્ટેટમેન્ટ વેરિએબલની કિંમતના આધારે બહુવિધ વિકલ્પોમાંથી એક કોડ બ્લોક પસંદ કરે છે.



```

1  #include <stdio.h>
2
3  int main() {
4      int choice;
5
6      printf("Enter operation (1-4): ");
7      scanf("%d", &choice);
8
9      switch(choice) {
10         case 1:
11             printf("Addition selected");
12             break;
13         case 2:
14             printf("Subtraction selected");
15             break;
16         case 3:
17             printf("Multiplication selected");
18             break;
19         case 4:
20             printf("Division selected");
21             break;
  
```

```

22     default:
23         printf("Invalid choice");
24     }
25
26     return 0;
27 }

```

- **Expression:** પૂર્ણાંક અથવા અક્ષર expression લે છે
- **Case labels:** અચળ expressions હોવા જોઈએ
- **Break statement:** આગલા કેસમાં પડતા અટકાવે છે
- **Default:** કોઈપણ કેસ સાથે મેચ ન થતી કિંમતો સંભાળે છે

મેમરી ટ્રીક

"SCBD - Switch Cases Break Default"

પ્રશ્ન 4 [a ગુણ]

3 Develop a C program to convert uppercase alphabet to lowercase alphabet.

જવાબ

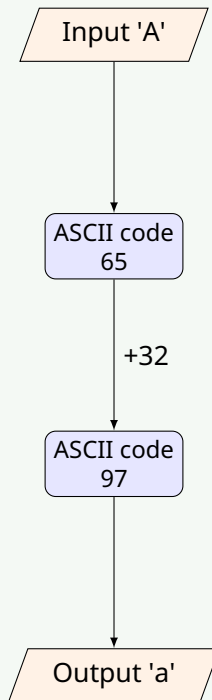
Answer:

```

1  #include <stdio.h>
2
3  int main() {
4      char upper, lower;
5
6      printf("Enter uppercase letter: ");
7      scanf("%c", &upper);
8
9      lower = upper + 32;
10     // Alternatively: lower = tolower(upper);
11
12     printf("Lowercase letter: %c", lower);
13
14     return 0;
15 }

```

Diagram:



મેમરી ટ્રીક

"ASCII-32 - Add 32 to Shift Characters Into Lowercase"

પ્રશ્ન 4 [b ગુણ]

4 What is pointer? Explain with example.

જવાબ

Answer:

પોઇન્ટર એ એક વેરિએબલ છે જે બીજા વેરિએબલનું મેમરી એડ્રેસ સ્ટોર કરે છે.

Concept	Syntax	Description
Declaration	int *p;	int પોઇન્ટર p ડિક્લેર કરે છે
Initialization	p = &var;	var નું એડ્રેસ p માં સ્ટોર કરે છે
Dereferencing	*p = 10;	પોઇન્ટર વેલ્યુ ને એક્સેસ/મોડિફાઇ કરે છે
Pointer arithmetic	p++	પછીના મેમરી લોકેશન પર જાય છે

```

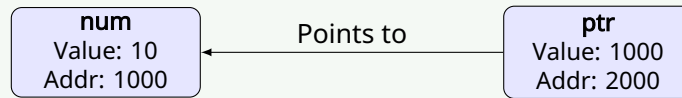
1 #include <stdio.h>
2
3 int main() {
4     int num = 10;
5     int *ptr;
6
7     ptr = &num; // Store address of num in ptr
8
9     printf("Value of num: %d\n", num);
10    printf("Address of num: %p\n", &num);
11    printf("Value of ptr: %p\n", ptr);
12    printf("Value pointed by ptr: %d\n", *ptr);
13
14    *ptr = 20; // Change value using pointer
  
```

```

15 printf("New value of num: %d\n", num);
16
17 return 0;
18 }

```

Memory Diagram:



મેમરી ટ્રીક

"SAID - Store Address to Indirectly Dereference"

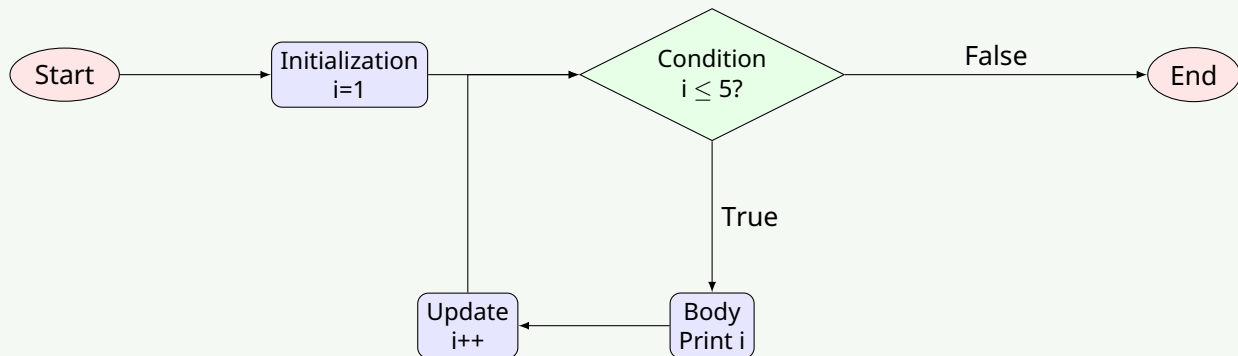
પ્રશ્ન 4 [c ગુણ]

7 Draw flowchart and explain for loop with example.

જવાબ

Answer:

for loop નો ઉપયોગ કોડ બ્લોકને ચોક્કસ સંખ્યામાં પુનરાવર્તિત કરવા માટે થાય છે.



```

1 #include <stdio.h>
2
3 int main() {
4     int i;
5
6     // Syntax: for(initialization; condition; update)
7     for(i = 1; i <= 5; i++) {
8         printf("%d ", i);
9     }
10    // Output: 1 2 3 4 5
11
12    return 0;
13 }

```

- **Initialization:** લૂપ શરૂ થતા પહેલાં એકવાર એક્ઝિક્યુટ થાય છે
- **Condition:** દરેક પુનરાવર્તન પહેલાં ચેક થાય છે
- **Update:** દરેક પુનરાવર્તન પછી એક્ઝિક્યુટ થાય છે
- **Body:** કોડ બ્લોક જે પુનરાવર્તિત થાય છે

મેમરી ટ્રીક

"ICU-B - Initialize, Check, Update, Body"

OR

પ્રશ્ન 4 [a ગુણ]

3 Develop a C program to find area of a triangle ($0.5 * \text{base} * \text{height}$)?

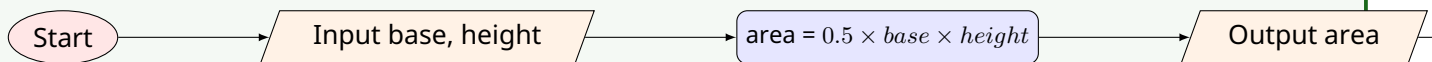
જવાબ

Answer:

```

1 #include <stdio.h>
2
3 int main() {
4     float base, height, area;
5
6     printf("Enter base of triangle: ");
7     scanf("%f", &base);
8
9     printf("Enter height of triangle: ");
10    scanf("%f", &height);
11
12    area = 0.5 * base * height;
13
14    printf("Area of triangle: %.2f", area);
15
16    return 0;
17 }
```

Diagram:



મેમરી ટ્રીક

"BHA - Base times Height divided by two equals Area"

OR

પ્રશ્ન 4 [b ગુણ]

4 Explain declaration and initialization of pointer.

જવાબ

Answer:

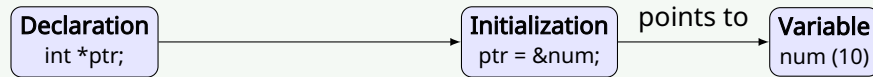
Operation	Syntax	Example	Description
Declaration	datatype *name;	int *ptr;	પોઇન્ટર વેરિએબલ બનાવે છે
Initialization	name = &var;	ptr = #	પોઇન્ટરને એડ્રેસ અસાઇન કરે છે
Combined	datatype *name = &var;	int *ptr = #	ઇનિશિયલાઇઝેશન સાથે ડેક્લેરેશન
NULL pointer	name = NULL;	ptr = NULL;	સેફ ઇનિશિયલાઇઝેશન

```

1 #include <stdio.h>
2
3 int main() {
4     int num = 10;    // Regular variable
5     int *ptr1;       // Declaration only
6     int *ptr2 = &num; // Declaration with initialization
7
8     ptr1 = &num;     // Initialization of ptr1
9
10    printf("num value: %d\n", num);
11    printf("num address: %p\n", &num);
12    printf("ptr1 value: %p\n", ptr1);
13    printf("ptr2 value: %p\n", ptr2);
14
15    return 0;
16 }

```

Diagram:



મેમરી ટ્રીક

"PAIN - Pointer Allocate, Initialize, Navigate"

OR

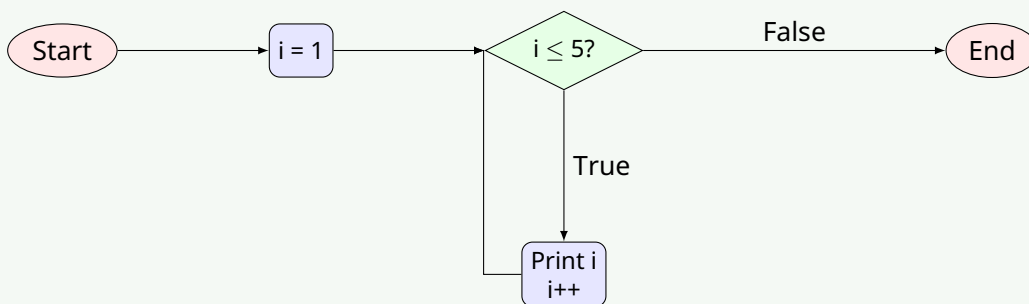
પ્રશ્ન 4 [C ગુણ]

7 Draw flowchart and explain while loop with example.

જવાબ

Answer:

જો શરત સાચી હોય ત્યાં સુધી while loop કોડ બ્લોકનું પુનરાવર્તન કરે છે.



```

1 #include <stdio.h>
2

```

```

3 int main() {
4     int i = 1;
5
6     // Syntax: while(condition)
7     while(i <= 5) {
8         printf("%d ", i);
9         i++;
10    }
11    // Output: 1 2 3 4 5
12
13    return 0;
14 }

```

- **Entry controlled:** એકિઝક્યુશન પહેલાં કન્ડિશન ચેક થાય છે
- **Infinite loop:** જો કન્ડિશન ક્યારેય ખોટી ન થાય
- **Components:** ઇનિશિયલાઇઝેશન, કન્ડિશન, બોડી, અપડેટ

મેમરી ટ્રીક

"WET - While Entry Test"

પ્રશ્ન 5 [a ગુણ]

૩ બુકની આપેલ માહિતી સ્ટોર કરવાનું સ્ટ્રક્ચર બનાવો: book_no, book_title, book_author, book_price

જવાબ

Answer:

```

1 #include <stdio.h>
2 #include <string.h>
3
4 struct Book {
5     int book_no;
6     char book_title[50];
7     char book_author[50];
8     float book_price;
9 };
10
11 int main() {
12     struct Book book1;
13
14     book1.book_no = 101;
15     strcpy(book1.book_title, "Programming in C");
16     strcpy(book1.book_author, "Dennis Ritchie");
17     book1.book_price = 450.50;
18
19     printf("Book No: %d\n", book1.book_no);
20     printf("Title: %s\n", book1.book_title);
21     printf("Author: %s\n", book1.book_author);
22     printf("Price: %.2f", book1.book_price);
23
24     return 0;
25 }

```

આકૃતિ:

struct Book

book_no (int)
book_title (char[50])
book_author (char[50])
book_price (float)

મેમરી ટ્રીક

"SNAP - Structure Needs All Properties"

પ્રશ્ન 5 [b ગુણ]

4 (1)sqrt() (2)pow() (3)strlen() (4)strcpy() ફંક્શનો ઉદાહરણ સાથે સમજાવો.

જવાબ**Answer:**

ફંક્શન	લાઇબ્રેરી	હેતુ	ઉદાહરણ
sqrt()	math.h	વર્ગમૂળ ગણો છે	sqrt(16) → 4.0
pow()	math.h	પાવર ગણો છે	pow(2, 3) → 8.0
strlen()	string.h	સ્ટ્રિંગની લંબાઈ શોધે છે	strlen("Hi") → 2
strcpy()	string.h	સ્ટ્રિંગની કોપી કરે છે	strcpy(d, s)

```

1 #include <stdio.h>
2 #include <math.h>
3 #include <string.h>
4
5 int main() {
6     double sqrtResult = sqrt(25);
7     double powResult = pow(2, 4);
8     char str[] = "Programming";
9     char dest[20];
10    int length = strlen(str);
11
12    strcpy(dest, str);
13
14    printf("sqrt(25) = %.2f\n", sqrtResult);
15    printf("pow(2, 4) = %.2f\n", powResult);
16    printf("Length of '%s' = %d\n", str, length);
17    printf("Copied string: %s\n", dest);
18
19    return 0;
20 }
```

મેમરી ટ્રીક

"SPSS - Square-root Power String-length String-copy"

પ્રશ્ન 5 [c ગુણ]

7 એરે અને એરેનું ઈનીશિયલાઈઝેશન ઉદાહરણ સાથે સમજાવો.

જવાબ

Answer:

એરે એ સમાન ડેટા એલિમેન્ટ્સનો સમૂહ છે જે સતત મેમરી લોકેશન પર સ્ટોર થાય છે.

પદ્ધતિ	સિન્ટેક્સ	ઉદાહરણ
ડેકલેરેશન	type name[size];	int marks[5];
ઇનિશિયલાઇઝેશન	type name[] = {v1,...};	int a[]={1,2};
એલિમેન્ટ એક્સેસ	name[index] = val;	marks[0]=90;

```

1 #include <stdio.h>
2
3 int main() {
4     // એરે ડેકલેરેશન અને ઇનિશિયલાઇઝેશન
5     int numbers[5] = {10, 20, 30, 40, 50};
6
7     // એરે એલિમેન્ટ્સ એક્સેસ કરવા
8     printf("First element: %d\n", numbers[0]);
9
10    numbers[1] = 25; // એરે એલિમેન્ટ બદલવી
11
12    printf("Array elements: ");
13    for(int i = 0; i < 5; i++) {
14        printf("%d ", numbers[i]);
15    }
16
17    return 0;
18 }
```

આકૃતિ:

10	20	30	40	50
0	1	2	3	4

મેમરી ટ્રીક

"CASED - Contiguous Arrangement of Similar Elements with Direct-access"

OR

પ્રશ્ન 5 [a ગુણ]

3 એરે અને સ્ટ્રક્ચર નો તફાવત લખો.

જવાબ

Answer:

ફીચર	એરે	સ્ટ્રક્ચર
ડેટા ટાઈપ્સ	માત્ર એક જ ડેટા ટાઈપ (સમાન)	અલગ અલગ ડેટા ટાઈપ્સ મંજૂર
એક્સેસ	ઇન્ડેક્સ દ્વારા: arr[0]	ડોટ ઓપરેટર દ્વારા: emp.id
મેમરી	સતત ફાળવણી	સતત ન પણ હોઈ શકે
હેતુ	સમાન આઈટમ્સનો સંગ્રહ	સંબંધિત આઈટમ્સનો સંગ્રહ

આકૃતિ:

Array [1, 2, 3]

Struct {id:1, name:'A'}

મેમરી ટ્રીક

"HASDIP - Homogeneous vs. Assorted, Same vs. Different, Index vs. Point"

OR

પ્રશ્ન 5 [b ગુણ]

4 યુઝર ડીફાઈન ફંક્શન એટલે શું? ઉદાહરણ સાથે સમજાવો.

જવાબ

Answer:

યુઝર-ડિફાઈનડ ફંક્શન એ એક કોડ બ્લોક છે જે ચોક્કસ કાર્ય કરે છે.

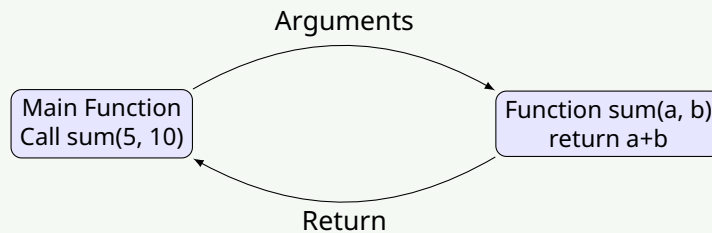
કોમ્પોનન્ટ	વિગત
રિટર્ન ટાઇપ	પરત કરવામાં આવતા ડેટાનો પ્રકાર (e.g., int)
ફંક્શન નામ	ઓળખકર્તા (e.g., sum)
પેરામીટર્સ	ઇનપુટ વેલ્યુઝ (e.g., int a, int b)
બોડી	એક્ઝિક્યુટ થતો કોડ

```

1 #include <stdio.h>
2
3 // ફંક્શન ડેક્લેરેશન
4 int sum(int a, int b);
5
6 int main() {
7     int res = sum(5, 10); // ફંક્શન કોલ
8     printf("Sum = %d", res);
9     return 0;
10 }
11
12 // ફંક્શન ડેફિનિશન
13 int sum(int a, int b) {
14     return a + b;
15 }

```

આકૃતિ:



મેમરી ટ્રીક

"CRPB - Create, Return, Pass, Body"

OR

પ્રશ્ન 5 [c ગુણ]

7 એરેના ઘટકોનો સરવાળો અને સરેરાશ શોધવા માટેનો C પ્રોગ્રામ લખો.

જવાબ

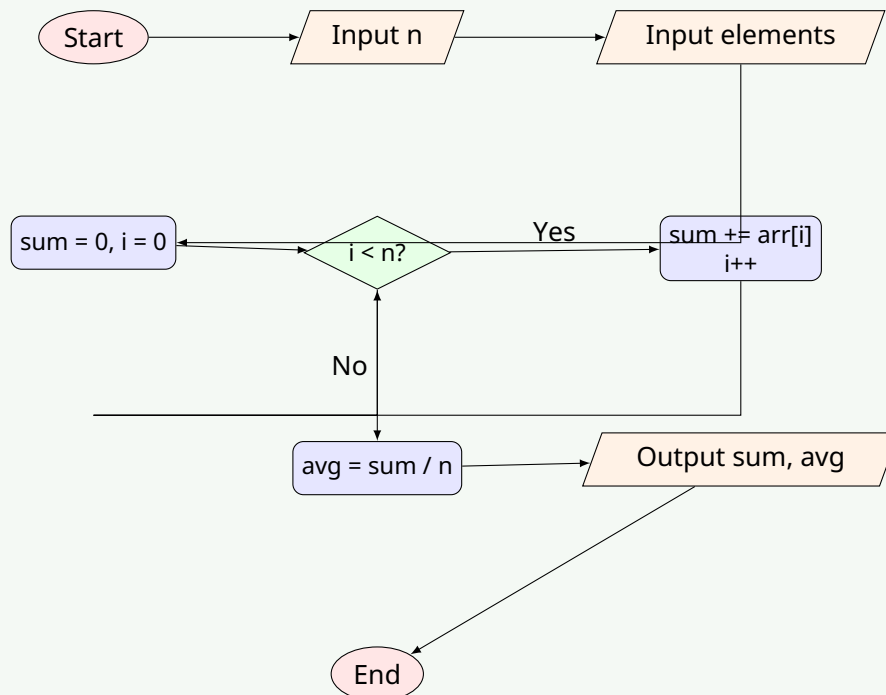
Answer:

```

1 #include <stdio.h>
2
3 int main() {
4     int arr[100], n, i;
5     int sum = 0;
6     float avg;
7
8     printf("Enter number of elements: ");
9     scanf("%d", &n);
10
11    printf("Enter %d elements:\n", n);
12    for(i = 0; i < n; i++) {
13        scanf("%d", &arr[i]);
14        sum += arr[i]; // દરેક એલેમિન્ટ ઉમેરો
15    }
16
17    avg = (float)sum / n; // સરેરાશ ગણતરી
18
19    printf("Sum: %d\n", sum);
20    printf("Avg: %.2f", avg);
21
22    return 0;
23 }

```

આકૃતિ:



મેમરી ટ્રીક

"LISA - Loop, Increment, Sum, Average"