

# Programming in C (4331105) - Summer 2025 Solution

Milav Dabgar

May 20, 2025

## પ્રશ્ન 1 [a ગુણ]

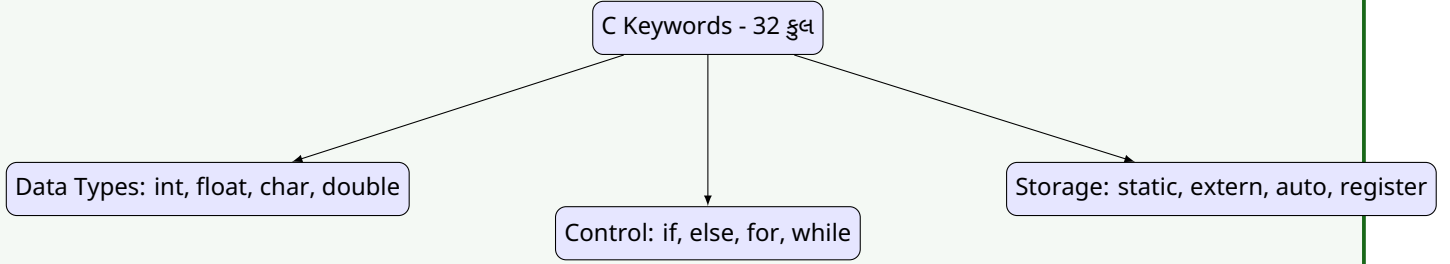
3 C માં કેટલા keywords છે? કોઈપણ ચાર keywords લખો

જવાબ

જવાબ:

કુલ Keywords	ઉદાહરણો
32 keywords	int, float, char, if

આકૃતિ:



- 32 keywords: C ભાષામાં કુલ આરક્ષિત શબ્દો
- Data type keywords: int, float, char, double વેરિએબલ જાહેર કરવા માટે
- Control keywords: if, else, for, while પ્રોગ્રામ ફ્લો માટે

### મેમરી ટ્રીક

"બિલાડી ચાર રંગમાં" (char, int, float, const)

## પ્રશ્ન 1 [b ગુણ]

4 વેરિએબલ શું છે? ઉદાહરણ સાથે વેરિએબલને નામ આપવાના નિયમો સમજાવો

જવાબ

જવાબ:

વેરિએબલ વ્યાખ્યા:

પાસાં	વર્ણન
વ્યાખ્યા	ડેટા સ્ટોર કરવા માટે નામવાળી મેમરી લોકેશન
હેતુ	પ્રોગ્રામ દરમિયાન બદલાતા મૂલ્યો રાખવા
જાહેરાત	datatype variable_name;

નામકરણના નિયમો:

- **પ્રથમ અક્ષર:** અક્ષર અથવા underscore (\_) હોવો જોઈએ
- **પછીના અક્ષરો:** અક્ષરો, અંકો, underscore માત્ર
- **Case sensitive:** 'Age' અને 'age' અલગ છે
- **Keywords નહીં:** 'int', 'float' જેવા આરક્ષિત શબ્દો વાપરી શકાતા નથી

ઉદાહરણો:

```
1 int age; // યોગ્ય
2 float _salary; // યોગ્ય
3 char name123; // યોગ્ય
4 int 2number; // ખોટું - અંકથી શરૂ
5 float for; // ખોટું - keyword વપરાયું
```

#### મેમરી ટ્રીક

"અક્ષર પહેલાં, keywords નહીં"

## પ્રશ્ન 1 [C ગુણ]

7 નીચેના statements માં errors શોધો

જવાબ

જવાબ:

Statement	Error	કારણ
(1) fLoat x;	ખોટો keyword	સાચું: float x;
(2) int min, max = 20;	અર્ધ initialization	માત્ર max initialize, min નહીં
(3) long char c;	ખોટું combination	long ને char સાથે વાપરી શકાતું નથી
(4) iNt a;	ખોટો keyword	સાચું: int a;
(5) FLOAT f=2;	ખોટો keyword	સાચું: float f=2;
(6) double m ; n;	Missing datatype	સાચું: double m, n;
(7) Int score (100)0;	અનેક errors	ખોટું syntax, સાચું: int score = 100;

મુખ્ય મુદ્દાઓ:

- **Case sensitivity:** Keywords નાના અક્ષરમાં હોવા જોઈએ
- **Multiple declaration:** Comma separator વાપરો
- **Initialization syntax:** = operator વાપરો

#### મેમરી ટ્રીક

"Keywords હંમેશા નાના અક્ષરે"

OR

## પ્રશ્ન 1 [C ગુણ]

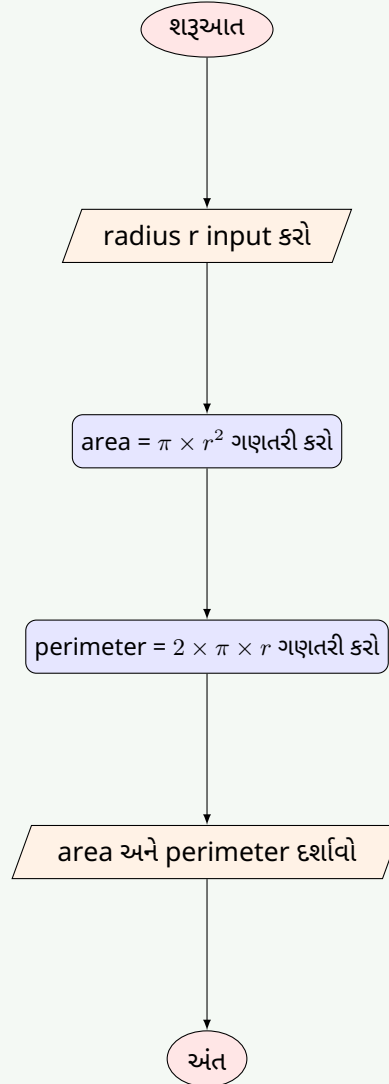
7 અલ્ગોરિથમ શું છે? ફ્લોચાર્ટ શું છે? વર્તુળનો વિસ્તાર અને પરિમિતિ શોધવા માટે ફ્લોચાર્ટ દોરો.

જવાબ

જવાબ:  
વ્યાખ્યાઓ:

શબ્દ	વ્યાખ્યા
Algorithm	સમસ્યા હલ કરવાની પગલાબદ્ધ પ્રક્રિયા
Flowchart	Algorithm નું પ્રતીકો વડે દ્રશ્ય પ્રતિનિધિત્વ

વર્તુળ માટે ફ્લોચાર્ટ:



Algorithm ના પગલાં:

- પગલું 1: શરૂઆત કરો
- પગલું 2: Radius નું મૂલ્ય input કરો
- પગલું 3:  $\pi \times r^2$  સૂત્ર વડે area ગણો
- પગલું 4:  $2 \times \pi \times r$  સૂત્ર વડે perimeter ગણો

મેમરી ટ્રીક

"શરૂ Input ગણતરી દર્શાવો અંત"

## પ્રશ્ન 2 [a ગુણ]

3 ઓપરેટર શું છે? C ના બધા operators ની યાદી બનાવો.

## જવાબ

જવાબ:

ઓપરેટર વ્યાખ્યા:

પાસાં	વર્ણન
વ્યાખ્યા	Operands પર operations કરતા ખાસ પ્રતીકો
હેતુ	ડેટા અને વેરિએબલ્સ સાથે કામ કરવા

C ઓપરેટર્સ યાદી:

વર્ગ	Operators
Arithmetic	+, -, *, /, %
Relational	<, >, <=, >=, ==, !=
Logical	&&,   , !
Assignment	=, +=, -=, *=, /=
Increment/Decrement	++, --
Conditional	?:

## મેમરી ટ્રીક

"ગણતરી, સંબંધ, તર્ક, અસાધન, વધારો, શરત"

## પ્રશ્ન 2 [b ગુણ]

4 while અને do while loop વચ્ચે તફાવત લખો.

## જવાબ

જવાબ:

પાસાં	while loop	do-while loop
Entry condition	Pre-tested	Post-tested
ન્યૂનતમ execution	0 વખત	ઓછામાં ઓછું 1 વખત
Syntax	while(condition) { }	do { } while(condition);
Semicolon	while પછી જરૂરી નથી	while પછી જરૂરી છે

ઉદાહરણ:

```

1 // while loop
2 while(i < 5) {
3     printf("%d", i);
4     i++;
5 }
6
7 // do-while loop
8 do {
9     printf("%d", i);
10    i++;
11 } while(i < 5);

```

મુખ્ય મુદ્દાઓ:

- **Pre-tested:** Execution પહેલાં condition ચકાસાય
- **Post-tested:** Execution પછી condition ચકાસાય

## મેમરી ટ્રીક

"While પહેલાં, Do પછી"

## પ્રશ્ન 2 [c ગુણ]

7 scanf() function નો formatted input માટે કેવી રીતે ઉપયોગ થાય છે? ઉદાહરણ સાથે સમજાવો

**જવાબ**

**જવાબ:**

**scanf() Function:**

લક્ષણ	વર્ણન
હેતુ	Keyboard થી formatted input વાંચવા
Syntax	scanf("format_string", &variable);
Return	સફળતાપૂર્વક વંચાયેલા inputs ની સંખ્યા

**Format Specifiers:**

Specifier	Data Type
%d	int
%f	float
%c	char
%s	string

**ઉદાહરણો:**

```
1 int age;
2 float salary;
3 char grade;
4
5 scanf("%d", &age); // Integer વાંચો
6 scanf("%f", &salary); // Float વાંચો
7 scanf("%c", &grade); // Character વાંચો
8 scanf("%d %f", &age, &salary); // બહુવધે inputs
```

**મહત્વના મુદ્દાઓ:**

- **Address operator (&):** Variables માટે જરૂરી
- **Format string:** Data types સાથે match થવું જોઈએ
- **Buffer issues:** જરૂર પડે તો fflush(stdin) વાપરો

**મેમરી ટ્રીક**

"Address Format Match"

OR

## પ્રશ્ન 2 [a ગુણ]

3 C ભાષાના arithmetic અને relational operators ની યાદી બનાવો

**જવાબ**

**જવાબ:**

Operator Type	Operators	હેતુ
Arithmetic	+, -, *, /, %	ગાણિતિક operations
Relational	<, >, <=, >=, ==, !=	Comparison operations

**ઉદાહરણો:**

```
1 // Arithmetic
2 int a = 10 + 5; // Addition
```

```
3 | int b = 10 % 3; // Modulus (remainder)
4 |
5 | // Relational
6 | if(a > b)      // મોટું
7 | if(a == b)    // બરાબર
```

મેમરી ટ્રીક

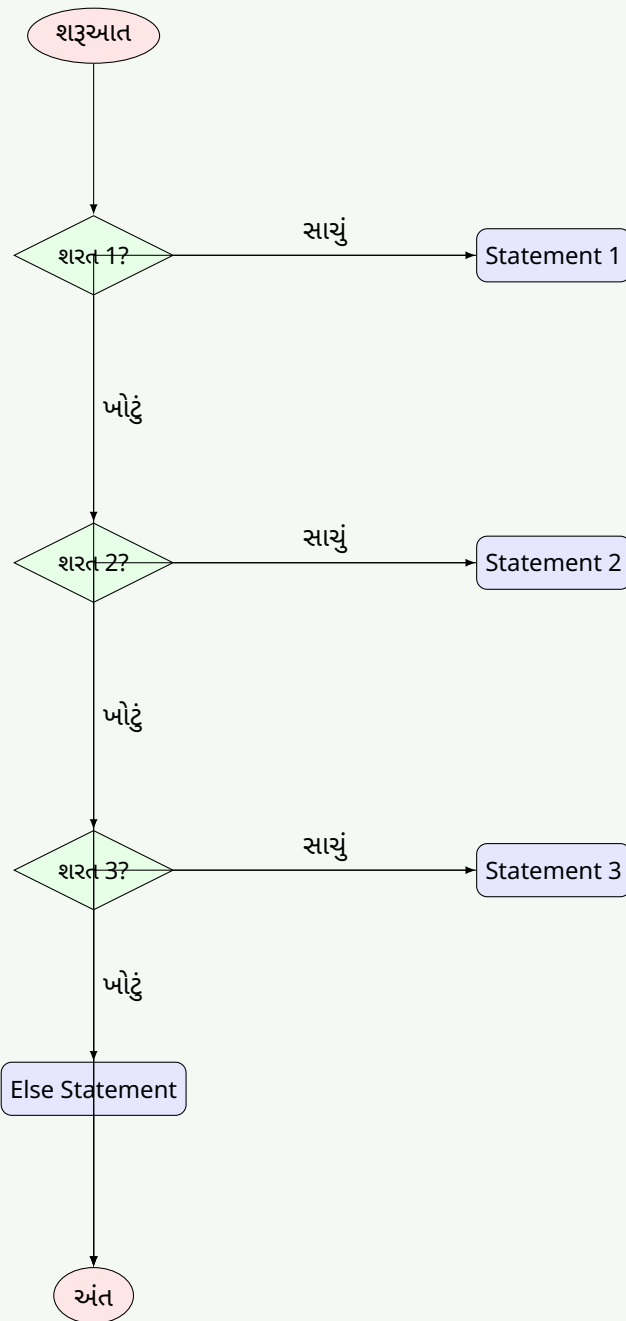
"ગણતરી સરખામણી"

## પ્રશ્ન 2 [b ગુણ]

4 else if ladder નો flow chart દોરો.

જવાબ

જવાબ:



માનખું:

- બહુવિધ શરતો: ક્રમમાં ચકાસાય છે
- પ્રથમ સાચી: તેનો block execute થાય
- Default case: કોઈ match ન મળે તો else block

મેમરી ટ્રીક

"પ્રથમ સાચી શોધો execute કરો"

## પ્રશ્ન 2 [c ગુણ]

7 printf() function નો formatted output માટે કેવી રીતે ઉપયોગ થાય છે? ઉદાહરણ સાથે સમજાવો

## જવાબ

જવાબ:

## printf() Function:

લક્ષણ	વર્ણન
હેતુ	Screen પર formatted output દર્શાવવા
Syntax	printf("format_string", variables);
Return	Print કરાયેલા characters ની સંખ્યા

## Format Specifiers:

Specifier	વપરાશ	ઉદાહરણ
%d	Integer	printf("%d", 25);
%f	Float	printf("%.2f", 3.14);
%c	Character	printf("%c", 'A');
%s	String	printf("%s", "Hello");

## Advanced Formatting:

```

1 int num = 123;
2 float pi = 3.14159;
3
4 printf("Number: %5d\n", num); // Width specification
5 printf("Pi: %.2f\n", pi); // Precision specification
6 printf("Hex: %x\n", num); // Hexadecimal
7 printf("Left aligned: %-10d\n", num); // Left alignment

```

## Escape Sequences:

- \n: નવી લીટી
- \t: Tab space
- \\: Backslash

## મેમરી ટ્રીક

"Format Width Precision Align"

## પ્રશ્ન 3 [a ગુણ]

3 Logical operators ની યાદી બનાવો અને તેને સમજાવો

## જવાબ

જવાબ:

Operator	Symbol	વર્ણન	Truth Table
AND	&&	બંને operands સાચા હોય તો સાચું	T&&T = T, બાકી = F
OR		કોઈપણ operand સાચો હોય તો સાચું	F  F = F, બાકી = T
NOT	!	Condition ઉલટાવે છે	!T = F, !F = T

## ઉદાહરણો:

```

1 int a = 5, b = 10;
2
3 if(a > 0 && b > 0) // બંને શરતો સાચી હોવી જોઈએ
4 if(a > 15 || b > 5) // ઓછામાં ઓછી એક શરત સાચી
5 if(!(a > 10)) // શરતનું નકારણ

```



## મેમરી ટ્રીક

"અને અથવા નહીં"

## પ્રશ્ન 3 [b ગુણ]

4 for loop ને ઉદાહરણ સાથે સમજાવો.

## જવાબ

જવાબ:

For Loop માળખું:

ઘટક	હેતુ
Initialization	શરુઆતી મૂલ્ય સેટ કરવું
Condition	ચાલુ રાખવા માટે ટેસ્ટ
Update	Loop variable ને બદલવું

Syntax:

```
1 for(initialization; condition; update) {
2     statements;
3 }
```

ઉદાહરણ:

```
1 // 1 થી 5 સુધીના નંબર print કરો
2 for(int i = 1; i <= 5; i++) {
3     printf("%d ", i);
4 }
5 // Output: 1 2 3 4 5
```

Execution Flow:

- પગલું 1: i = 1 initialize કરો
- પગલું 2: i <= 5 condition ચકાસો
- પગલું 3: Statements execute કરો
- પગલું 4: i++ update કરો, પગલું 2 પર પાછા

## મેમરી ટ્રીક

"Initialize ચકાસો Execute Update"

## પ્રશ્ન 3 [c ગુણ]

7 ત્રણ પૂર્ણાંક સંખ્યાઓ x અને y માંથી મહત્તમ શોધવા માટે પ્રોગ્રામ લખો.

## જવાબ

જવાબ:

```
1 #include <stdio.h>
2
3 int main() {
4     int x, y, z, max;
5 }
```

```

6   printf("ત્રણ સંખ્યાઓ દાખલ કરો: ");
7   scanf("%d %d %d", &x, &y, &z);
8
9   max = x; // પ્રથમ સંખ્યાને મહત્તમ માનો
10
11  if(y > max) {
12      max = y;
13  }
14  if(z > max) {
15      max = z;
16  }
17
18  printf("મહત્તમ સંખ્યા છે: %d", max);
19
20  return 0;
21  }

```

#### Algorithm ના પગલાં:

પગલું	કાર્ય
1	ત્રણ સંખ્યાઓ input કરો
2	પ્રથમને મહત્તમ માનો
3	બીજી સાથે સરખામણી, મોટી હોય તો update
4	ત્રીજી સાથે સરખામણી, મોટી હોય તો update
5	મહત્તમ દર્શાવો

#### વૈકલ્પિક પદ્ધતિ:

```
1 max = (x > y) ? ((x > z) ? x : z) : ((y > z) ? y : z);
```

#### મેમરી ટ્રીક

"માનો સરખાવો Update દર્શાવો"

OR

## પ્રશ્ન 3 [a ગુણ]

3 conditional operator ને ઉદાહરણ સાથે સમજાવો.

#### જવાબ

જવાબ:

#### Conditional Operator (Ternary):

લક્ષણ	વર્ણન
Symbol	?:
Syntax	condition ? value1 : value2
હેતુ	if-else નો ટૂંકો રસ્તો

#### ઉદાહરણો:

```

1  int a = 10, b = 20;
2  int max = (a > b) ? a : b;    // max = 20
3
4  char grade = (marks >= 60) ? 'P' : 'F';
5  printf("Status: %s", (age >= 18) ? "Adult" : "Minor");

```

## સમાન if-else:

```

1 if(a > b)
2     max = a;
3 else
4     max = b;

```

## ફાયદાઓ:

- સંક્ષિપ્ત: એક લીટીમાં expression
- કાર્યક્ષમ: ઝડપી execution

## મેમરી ટ્રીક

"પ્રશ્નચિહ્ન કોલન પસંદગી"

## પ્રશ્ન 3 [b ગુણ]

4 while loop ને ઉદાહરણ સાથે સમજાવો.

## જવાબ

## જવાબ:

## While Loop:

લક્ષણ	વર્ણન
પ્રકાર	Entry-controlled loop
Syntax	while(condition) { statements; }
Execution	શરત સાચી હોય ત્યાં સુધી repeat

## ઉદાહરણ:

```

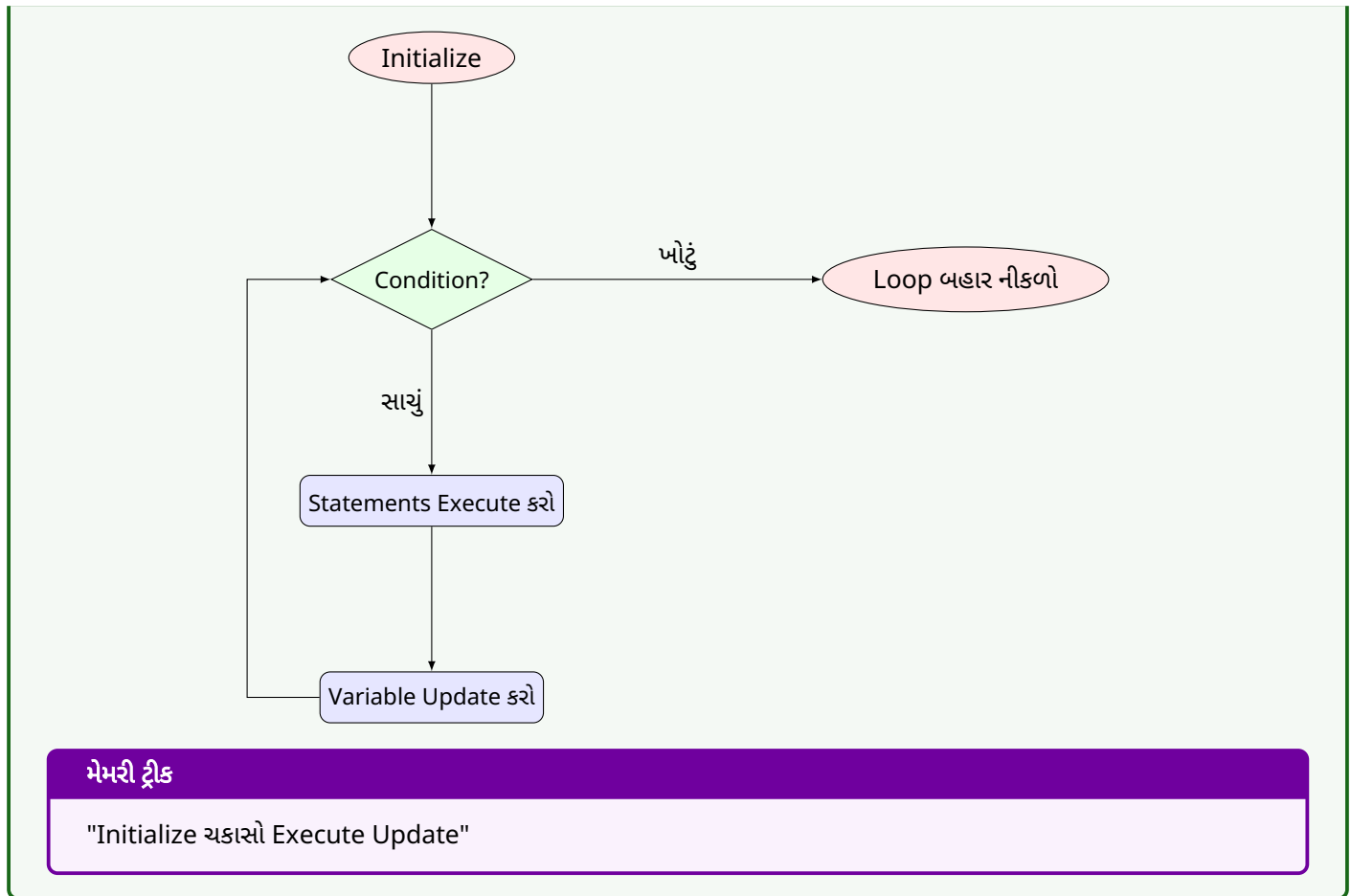
1 int i = 1;
2 while(i <= 5) {
3     printf("%d ", i);
4     i++;
5 }
6 // Output: 1 2 3 4 5

```

## મહત્વના મુદ્દાઓ:

- Initialization: Loop પહેલાં
- Condition: શરૂઆતમાં ચકાસાય
- Update: Loop body અંદર
- Infinite loop: જો condition ક્યારેય false ન બને

## Flowchart માળખું:



### પ્રશ્ન 3 [c ગુણ]

7 કીબોર્ડમાંથી પૂર્ણાંક વાંચવા માટે અને આપેલ સંખ્યા odd હોય કે even હોય તે શોધવા માટેનો પ્રોગ્રામ લખો.

જવાબ

જવાબ:

```

1 #include <stdio.h>
2
3 int main() {
4     int number;
5
6     printf("એક પૂર્ણાંક દાખલ કરો: ");
7     scanf("%d", &number);
8
9     if(number % 2 == 0) {
10        printf("%d એ સમ સંખ્યા છે", number);
11    }
12    else {
13        printf("%d એ વધિમ સંખ્યા છે", number);
14    }
15
16    return 0;
17 }

```

તર્ક સમજૂતી:

ખ્યાલ	વર્ણન
Modulus operator (%)	ભાગાકાર પછી બાકી આપે છે
સમ શરત	number % 2 == 0
વિષમ શરત	number % 2 != 0

વૈકલ્પિક પદ્ધતિઓ:

```

1 // પદ્ધતિ2: Conditional operator વાપરીને
2 printf("%d એ %s છે", number, (number % 2 == 0) ? "સમ" : "વિષમ");
3
4 // પદ્ધતિ3: Bitwise AND વાપરીને
5 if(number & 1)
6     printf("વિષમ");
7 else
8     printf("સમ");

```

Sample Output:

```

1 એક
2 પૂરણાંક દાખલ કરો: 7
3 7 એ વિષમ સંખ્યા છે

```

મેમરી ટ્રીક

"Modulus બે શૂન્ય સમ"

## પ્રશ્ન 4 [a ગુણ]

3 નીચેના arithmetic expressions નું મૂલ્યાંકન કરો:  $30/4*4 - 20\%6 + 17/2$

જવાબ

જવાબ:

પગલાબદ્ધ મૂલ્યાંકન:

પગલું	Expression	ગણતરી	પરિણામ
1	$30/4*4$	$(30/4)*4 = 7*4$	28
2	$20\%6$	$20 \bmod 6$	2
3	$17/2$	Integer division	8
4	અંતિમ	$28 - 2 + 8$	34

Operator પ્રાધાન્યતા:

પ્રાધાન્યતા	Operators
ઉચ્ચ	*, /, % (ડાબેથી જમણે)
નીચી	+, - (ડાબેથી જમણે)

સંપૂર્ણ ગણતરી:

```

1 30/4*4 - 20%6 + 17/2
2 = 7*4 - 2 + 8    // પહેલાં division અને modulus
3 = 28 - 2 + 8    // Multiplication
4 = 26 + 8        // +, - માટે ડાબેથી જમણે
5 = 34            // અંતિમ જવાબ

```

## મેમરી ટ્રીક

"ગુણા ભાગ પહેલાં બાદબાકી પછી"

## પ્રશ્ન 4 [b ગુણ]

4 5 પૂર્ણાંક સંખ્યાઓની array ની સરવાળો અને સરેરાશ શોધવા માટેનો પ્રોગ્રામ લખો.

## જવાબ

## જવાબ:

```

1 #include <stdio.h>
2
3 int main() {
4     int numbers[5];
5     int sum = 0;
6     float average;
7
8     printf("5 પૂર્ણાંકો દાખલ કરો:\n");
9     for(int i = 0; i < 5; i++) {
10         scanf("%d", &numbers[i]);
11         sum += numbers[i];
12     }
13
14     average = (float)sum / 5;
15
16     printf("સરવાળો = %d\n", sum);
17     printf("સરેરાશ = %.2f", average);
18
19     return 0;
20 }
```

## Algorithm:

1. 5 integers ની array જાહેર કરો
2. Sum ને 0 થી initialize કરો
3. Loop વાપરીને 5 numbers input કરો
4. દરેક number ને sum માં ઉમેરો
5. Average = sum/5 ગણો
6. પરિણામો દર્શાવો

## મુખ્ય મુદ્દાઓ:

- **Type casting:** (float)sum ચોક્કસ division માટે
- **Loop વપરાશ:** Repetitive input માટે કાર્યક્ષમ

## મેમરી ટ્રીક

"જાહેર Input ઉમેરો ગણો દર્શાવો"

## પ્રશ્ન 4 [c ગુણ]

7 Pointer વ્યાખ્યાયિત કરો. Pointers કેવી રીતે declared અને initialized કરવામાં આવે છે તે ઉદાહરણ સાથે સમજાવો.

## જવાબ

જવાબ:

Pointer વ્યાખ્યા:

પાસાં	વર્ણન
વ્યાખ્યા	બીજા variable નું memory address સ્ટોર કરતું variable
હેતુ	સીધી memory access અને dynamic memory allocation
Symbol	* (asterisk) declaration અને dereferencing માટે

Declaration અને Initialization:

```

1 // Declaration
2 int *ptr;    // Integer નો pointer
3 float *fptr; // Float નો pointer
4 char *cptr;  // Character નો pointer
5
6 // Initialization
7 int num = 10;
8 int *ptr = &num; // num ના address સાથે initialize
9
10 // વૈકલ્પિક
11 int *ptr;
12 ptr = &num; // પછીથી address assign

```

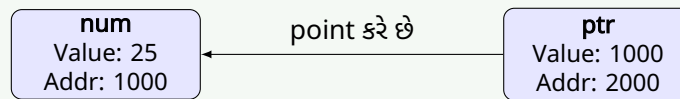
ઉદાહરણ પ્રોગ્રામ:

```

1 #include <stdio.h>
2
3 int main() {
4     int num = 25;
5     int *ptr = &num;
6
7     printf("num નું મૂલ્ય: %d\n", num);
8     printf("num નું address: %p\n", &num);
9     printf("ptr નું મૂલ્ય: %p\n", ptr);
10    printf("ptr દ્વારા pointed મૂલ્ય: %d\n", *ptr);
11
12    return 0;
13 }

```

Memory Diagram:



મેમરી ટ્રીક

"Address Star Dereference"

OR

## પ્રશ્ન 4 [a ગુણ]

3 નીચેના arithmetic expressions નું મૂલ્યાંકન કરો:  $50 / 3 \% 3 + 5 * 7$

## જવાબ

જવાબ:

પગલાબદ્ધ મૂલ્યાંકન:

પગલું	Expression	ગણતરી	પરિણામ
1	50/3	Integer division	16
2	16%3	16 mod 3	1
3	5*7	Multiplication	35
4	અંતિમ	1 + 35	36

સંપૂર્ણ ગણતરી:

```

1 50 / 3 % 3 + 5 * 7
2 = 16 % 3 + 35 // પહેલાં division અને multiplication
3 = 1 + 35 // Modulus operation
4 = 36 // અંતિમ જવાબ

```

Operator પ્રાધાન્યતા લાગુ:

- ઉચ્ચ પ્રાધાન્યતા: /, %, \* (ડાબેથી જમણે)
- નીચી પ્રાધાન્યતા: + (ડાબેથી જમણે)

## મેમરી ટ્રીક

"ભાગ Mod ગુણા ઉમેરો"

## પ્રશ્ન 4 [b ગુણ]

4 N પૂર્ણાંકોની array માં સૌથી મોટી સંખ્યા શોધવા માટેનો પ્રોગ્રામ લખો.

## જવાબ

જવાબ:

```

1 #include <stdio.h>
2
3 int main() {
4     int n, i;
5     int largest;
6
7     printf("elements ની સંખ્યા દાખલ કરો: ");
8     scanf("%d", &n);
9
10    int arr[n];
11
12    printf("%d સંખ્યાઓ દાખલ કરો:\n", n);
13    for(i = 0; i < n; i++) {
14        scanf("%d", &arr[i]);
15    }
16
17    largest = arr[0]; // પ્રથમ element ને largest માનો
18
19    for(i = 1; i < n; i++) {
20        if(arr[i] > largest) {
21            largest = arr[i];
22        }
23    }
24
25    printf("સૌથી મોટી સંખ્યા છે: %d", largest);
26

```



```

27     return 0;
28 }

```

**Algorithm:**

1. Array નું size input કરો
2. Array elements input કરો
3. પ્રથમ element ને largest માનો
4. બાકીના elements સાથે સરખામણી કરો
5. જો મોટું મળે તો largest update કરો
6. પરિણામ દર્શાવો

**મેમરી ટ્રીક**

"Input માનો સરખાવો Update દર્શાવો"

**પ્રશ્ન 4 [C ગુણ]**

7 Array વ્યાખ્યાયિત કરો. Array variable ની જરૂરિયાત સમજાવો. 1-D array ને ઉદાહરણ સાથે સમજાવો

**જવાબ**

**જવાબ:**

**Array વ્યાખ્યા:**

પાસાં	વર્ણન
વ્યાખ્યા	સમાન data type ના elements નો સંગ્રહ
Storage	સતત memory locations માં
Access	Index/subscript વાપરીને

**Arrays ની જરૂરિયાત:**

સમસ્યા	Array સાથે ઉકેલ
બહુવિધ values સ્ટોર કરવા	એક જ array variable
ઘણા variables ટાળવા	arr[100] બદલે a1, a2, ..., a100
કાર્યક્ષમ processing	Loop-based operations
Memory organization	Contiguous allocation

**1-D Array Declaration:**

```

1  datatype arrayname[size];
2
3  // ઉદાહરણો
4  int marks[5];      // 5 integers ની Array
5  float prices[10];  // 10 floats ની Array
6  char name[20];     // 20 characters ની Array

```

**Array Initialization:**

```

1  // પદ્ધતિ 1: Declaration વખતે
2  int numbers[5] = {10, 20, 30, 40, 50};
3
4  // પદ્ધતિ 2: વ્યક્તિગત assignment
5  int arr[3];
6  arr[0] = 5;
7  arr[1] = 15;
8  arr[2] = 25;

```

**ઉદાહરણ પ્રોગ્રામ:**

```

1 #include <stdio.h>
2
3 int main() {
4     int marks[5] = {85, 90, 78, 92, 88};
5     int i, sum = 0;
6
7     printf("વહિયાત્થીના ગુણ:\n");
8     for(i = 0; i < 5; i++) {
9         printf("વર્ષિય %d: %d\n", i+1, marks[i]);
10        sum += marks[i];
11    }
12
13    printf("કુલ ગુણ: %d", sum);
14    return 0;
15 }

```

### Memory Layout:

1000	1004	1008	1012	1016
85	90	78	92	88
marks[0]	marks[1]	marks[2]	marks[3]	marks[4]

### મેમરી ટ્રીક

"સમાન ડેટા સતત Index"

## પ્રશ્ન 5 [a ગુણ]

3 if ... else statement નું ઉદાહરણ આપો.

### જવાબ

#### જવાબ:

```

1 #include <stdio.h>
2
3 int main() {
4     int age;
5
6     printf("તમારી ઉંમર દાખલ કરો: ");
7     scanf("%d", &age);
8
9     if(age >= 18) {
10        printf("તમે મતદાન માટે લાયક છો");
11    }
12    else {
13        printf("તમે મતદાન માટે લાયક નથી");
14    }
15
16    return 0;
17 }

```

#### માનખું:

ઘટક	હેતુ
if	શરત ટેસ્ટ કરે છે
condition	Boolean expression
if-block	શરત સાચી હોય ત્યારે execute
else-block	શરત ખોટી હોય ત્યારે execute

#### Sample Outputs:

- Input: 20    Output: તમે મતદાન માટે લાયક છો
- Input: 16    Output: તમે મતદાન માટે લાયક નથી

#### મેમરી ટ્રીક

"જો સાચું નહીંતર ખોટું"

## પ્રશ્ન 5 [b ગુણ]

4 આપેલ character ની category ચકાસવા માટેનો પ્રોગ્રામ લખો.

#### જવાબ

##### જવાબ:

```

1 #include <stdio.h>
2 #include <ctype.h>
3
4 int main() {
5     char ch;
6
7     printf("એક character દાખલ કરો: ");
8     scanf("%c", &ch);
9
10    if(isdigit(ch)) {
11        printf("%c એ અંક છે", ch);
12    }
13    else if(isupper(ch)) {
14        printf("%c એ મોટા અક્ષર છે", ch);
15    }
16    else if(islower(ch)) {
17        printf("%c એ નાના અક્ષર છે", ch);
18    }
19    else {
20        printf("%c એ વર્ણિત પ્રતીક છે", ch);
21    }
22
23    return 0;
24 }
```

#### Character Categories:

Function	વર્ગ	Range
isdigit()	અંક	0-9
isupper()	મોટા અક્ષર	A-Z
islower()	નાના અક્ષર	a-z
Others	વિશેષ પ્રતીકો	!@#\$%^&* etc.

#### વૈકલ્પિક પદ્ધતિ:

```

1 if(ch >= '0' && ch <= '9')
2     printf("અંક");
3 else if(ch >= 'A' && ch <= 'Z')
4     printf("મોટા અક્ષર");
5 else if(ch >= 'a' && ch <= 'z')
6     printf("નાના અક્ષર");
7 else
8     printf("વશિષ્ટ પ્રતીક");

```

### મેમરી ટ્રીક

"અંક મોટા નાના વિશેષ"

## પ્રશ્ન 5 [c ગુણ]

7 Structure શું છે? તેની syntax યોગ્ય ઉદાહરણ સાથે સમજાવો

### જવાબ

જવાબ:

Structure વ્યાખ્યા:

પાસાં	વર્ણન
વ્યાખ્યા	વિવિધ data types ને જોડીને બનાવેલ user-defined data type
હેતુ	સંબંધિત data ને એક જ નામ હેઠળ જૂથ બનાવવા
Keyword	struct

Syntax:

```

1 struct structure_name {
2     datatype member1;
3     datatype member2;
4     ...
5 };

```

ઉદાહરણ - Student Structure:

```

1 #include <stdio.h>
2
3 struct Student {
4     int roll_no;
5     char name[50];
6     float marks;
7     char grade;
8 };
9
10 int main() {
11     struct Student s1;
12
13     // Data input
14     printf("રોલ નંબર દાખલ કરો: ");
15     scanf("%d", &s1.roll_no);
16
17     printf("નામ દાખલ કરો: ");
18     scanf("%s", s1.name);
19
20     printf("ગુણ દાખલ કરો: ");
21     scanf("%f", &s1.marks);

```

```

22
23 printf("ગ્રેડ દાખલ કરો: ");
24 scanf(" %c", &s1.grade);
25
26 // Data display
27 printf("\nવહિયાત્થીનીની વગિતો:\n");
28 printf("રોલ નં: %d\n", s1.roll_no);
29 printf("નામ: %s\n", s1.name);
30 printf("ગુણ: %.2f\n", s1.marks);
31 printf("ગ્રેડ: %c\n", s1.grade);
32
33 return 0;
34 }

```

### Structure લક્ષણો:

- **Dot operator (.):** Structure members ને access કરવા
- **Memory allocation:** કુલ size = બધા members નો સરવાળો
- **Initialization:** Declaration વખતે initialize કરી શકાય

### Structure Initialization:

```

1 struct Student s1 = {101, "John", 85.5, 'A'};

```

### Memory Layout:

#### s1 Structure

roll\_no (4 bytes)  
name (50 bytes)  
marks (4 bytes)  
grade (1 byte)

#### મેમરી ટ્રીક

"સંબંધિત ડેટાને એકસાથે જૂથ બનાવો"

OR

## પ્રશ્ન 5 [a ગુણ]

3 -5 અને +5 વચ્ચેના બધા numbers print કરવા માટેનો પ્રોગ્રામ લખો.

### જવાબ

#### જવાબ:

```

1 #include <stdio.h>
2
3 int main() {
4     int i;
5
6     printf("-5 અને +5 વચ્ચેના નંબરો:\n");
7
8     for(i = -5; i <= 5; i++) {
9         printf("%d ", i);
10    }
11
12    return 0;
13 }

```

**Output:**

```

1 -5 અને +5 વચ્ચેના નંબરો:
2 -5 -4 -3 -2 -1 0 1 2 3 4 5

```

**વૈકલ્પિક પદ્ધતિઓ:**

```

1 // પદ્ધતિ2: while loop વાપરીને
2 int i = -5;
3 while(i <= 5) {
4     printf("%d ", i);
5     i++;
6 }
7
8 // પદ્ધતિ3: બે અલગ loops
9 for(i = -5; i < 0; i++)
10     printf("%d ", i);
11 printf("0 ");
12 for(i = 1; i <= 5; i++)
13     printf("%d ", i);

```

**મેમરી ટ્રીક**

"નકારાત્મકથી શરૂ સકારાત્મકે અંત"

**પ્રશ્ન 5 [b ગુણ]**

4 Quadratic equation ના roots શોધવા માટેનો 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("coefficients (a, b, c) દાખલ કરો: ");
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 વાસ્તવિક અને અલગ છે\n");
17         printf("Root1 = %.2f\n", root1);
18         printf("Root2 = %.2f\n", root2);
19     }
20     else if(discriminant == 0) {
21         root1 = -b / (2*a);
22         printf("Roots વાસ્તવિક અને સમાન છે\n");
23         printf("Root = %.2f\n", root1);
24     }

```

```

25 else {
26     float realPart = -b / (2*a);
27     float imagPart = sqrt(-discriminant) / (2*a);
28     printf("Roots સંકુલ છે\n");
29     printf("Root1 = %.2f + %.2fi\n", realPart, imagPart);
30     printf("Root2 = %.2f - %.2fi\n", realPart, imagPart);
31 }
32
33 return 0;
34 }

```

### Quadratic Formula વિશ્લેષણ:

Discriminant	Roots નો પ્રકાર
$b^2 - 4ac > 0$	વાસ્તવિક અને અલગ
$b^2 - 4ac = 0$	વાસ્તવિક અને સમાન
$b^2 - 4ac < 0$	સંકુલ (કાલ્પનિક)

સૂત્ર:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

### મેમરી ટ્રીક

"Discriminant Roots નો પ્રકાર નક્કી કરે છે"

## પ્રશ્ન 5 [c ગુણ]

7 નીચેના built-in functions ને ઉદાહરણો સાથે સમજાવો

### જવાબ

જવાબ:

Function સમજૂતીઓ:

Function	હેતુ	Header File	ઉદાહરણ
clrscr()	Screen સાફ કરવા	conio.h	clrscr();
sqrt()	વર્ગમૂળ	math.h	sqrt(16) = 4.0
strlen()	String ની લંબાઈ	string.h	strlen("Hello") = 5
isdigit()	અંક છે કે કેમ ચકાસવા	ctype.h	isdigit('5') = true
isalpha()	અક્ષર છે કે કેમ ચકાસવા	ctype.h	isalpha('A') = true
toupper()	મોટા અક્ષરમાં બદલવા	ctype.h	toupper('a') = 'A'
tolower()	નાના અક્ષરમાં બદલવા	ctype.h	tolower('B') = 'b'

ઉદાહરણ પ્રોગ્રામ:

```

1 #include <stdio.h>
2 #include <math.h>
3 #include <string.h>
4 #include <ctype.h>
5
6 int main() {
7     // clrscr(); // Not standard in modern compilers
8
9     // sqrt() ઉદાહરણ
10    float num = 25.0;
11    printf("%.1f નું વર્ગમૂળ = %.2f\n", num, sqrt(num));
12
13    // strlen() ઉદાહરણ
14    char str[] = "Programming";

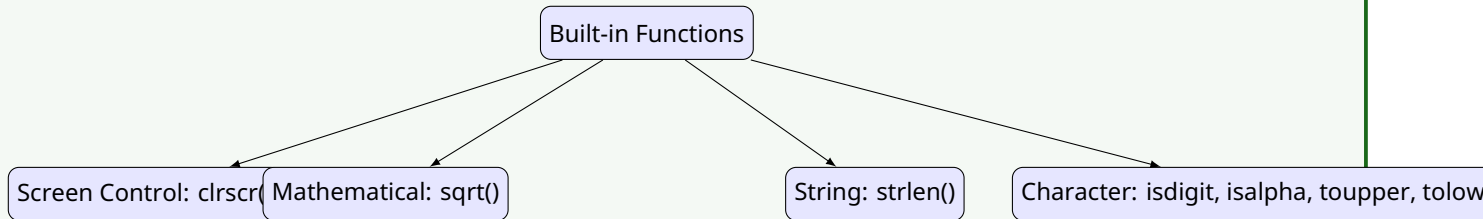
```

```

15 printf("%s' ની લંબાઈ = %d\n", str, strlen(str));
16
17 // Character functions
18 char ch = 'a';
19 printf("%c' અંક છે: %s\n", ch, isdigit(ch) ? "હા" : "ના");
20 printf("%c' અક્ષર છે: %s\n", ch, isalpha(ch) ? "હા" : "ના");
21 printf("toupper('%c') = '%c'\n", ch, toupper(ch));
22
23 return 0;
24 }

```

### Function વર્ગીકરણ:



### મેમરી ટ્રીક

"સાફ ગણિત String Character"