

Subject Name (Gujarati)

4331105 -- Summer 2025

Semester 1 Study Material

Detailed Solutions and Explanations

પ્રશ્ન 1(a) [3 ગુણ]

C માં કેટલા keywords છે? કોઈપણ ચાર keywords લખો

જવાબ

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

આફ્ટિસ:

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph TD
    A[C Keywords {- 32 } {-}{-}{-}] --- B[Data Types: int, float, char, double]
    A --- C[Control: if, else, for, while]
    A --- D[Storage: static, extern, auto, register]
{Highlighting}
{Shaded}



- 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 ફલો માટે

```

મેમરી ટ્રીક

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

પ્રશ્ન 1(b) [4 ગુણ]

વેરિએબલ શું છે? ઉદાહરણ સાથે વેરિએબલને નામ આપવાના નિયમો સમજાવો

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 (_) હોવા 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:

```
int age;      //
float \_salary;  //
char name123;   //
int 2number;    // {-
float for;     // {- keyword }
```

મેમરી ટ્રીક

"અક્ષર પહેલાં, keywords નહીં"

પ્રશ્ન 1(c) [7 ગુણ]

નીચેના statements માં errors શોધો

જવાબ

Statement	Error	કારણ
(1) fLoat x;	ખોટો keyword	સાચું: float x;
(2) int min, max = 20;	અધ્ય 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 હેઠળ નાના અક્ષરે"

પ્રશ્ન 1(c) OR [7 ગુણ]

અણોરિધમ શું છે? ફ્લોયાર્ડ શું છે? વર્ટુનો વિસ્તાર અને પરિમિતિ શોધવા માટે ફ્લોયાર્ડ દોરો.

જવાબ

વ્યાખ્યાઓ:

શબ્દ	વ્યાખ્યા
Algorithm	સમર્થા હલ કરવાની પગલાભદ્ર પ્રક્રિયા
Flowchart	Algorithm નું પ્રતીકો વડે દ્રશ્ય પ્રતિનિધિત્વ

વર્તુળ માટે ફ્લોચાર્ટ:

```
flowchart LR
    A[ ] --> B["B[radius r input ]"]
    B --> C["C[area = r^2 ]"]
    C --> D["D[perimeter = 2 r ]"]
    D --> E["E[area perimeter ]"]
    E --> F[ ]
```

Algorithm ના પગલાં:

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

મેમરી ટ્રીક

“શરૂ Input ગણતરી દર્શાવો અંત”

પ્રશ્ન 2(a) [3 ગુણ]

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

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 અસાઇન વધારો શરત”

પ્રશ્ન 2(b) [4 ગુણ]

while અને do while loop વર્ચે તફાવત લખો.

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

ઉદાહરણાઃ

```
// while loop
while(i {} 5) \{
    printf("\%d", i);
    i++;
\}

// do{-while loop  }
do \{
    printf("\%d", i);
    i++;
\} 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) [7 ગુણ]

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

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:

```
int age;
float salary;
char grade;

scanf("\%d", \&age);           // Integer
scanf("\%f", \&salary);        // Float
scanf("\%c", \&grade);         // Character
scanf("\%d \%f", \&age, \&salary); // inputs
```

મહત્વના મુદ્દાઓ:

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

મેમરી ટ્રીક

"Address Format Match"

પ્રશ્ન 2(a) OR [3 ગુણ]

C ભાષાના arithmetic અને relational operators ની યાદી બનાવો

જવાબ

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

ઉદાહરણો:

```
// Arithmetic
int a = 10 + 5;    // Addition
int b = 10 \% 3;   // Modulus (remainder)

// Relational
if(a {} b)          //
if(a == b)          //
```

મેમરી ટ્રીક

"ગણિતરી સરખામણી"

પ્રશ્ન 2(b) OR [4 ગુણ]

else if ladder નો flow chart દરે.

જવાબ

```
flowchart LR
    A[ ] --{-{-}}--> B{ 1? \}
    B --{-{-}}--> C[Statement 1]
    B --{-{-}}--> D{ 2? \}
    D --{-{-}}--> E[Statement 2]
    D --{-{-}}--> F{ 3? \}
    F --{-{-}}--> G[Statement 3]
    F --{-{-}}--> H[Else Statement]
    C --{-{-}}--> I[ ]
    E --{-{-}}--> I
    G --{-{-}}--> I
    H --{-{-}}--> I
```

માણસું:

- બહુવિધ શરતો: કમમાં 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) OR [7 ગુણ]

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

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:

```
int num = 123;
float pi = 3.14159;

printf("Number: \%5d{n}", num);           // Width specification
printf("Pi: \% .2f{n}", pi);             // Precision specification
printf("Hex: \%x{n}", num);              // Hexadecimal
printf("Left aligned: \%-10dn", num);    // Left alignment
```

Escape Sequences:

- ****: નવી લીટી
- ***: Tab space
- \: Backslash

મેમરી ટ્રીક

“Format Width Precision Align”

પ્રશ્ન 3(a) [3 ગુણ]

Logical operators ની યાદી બનાવો અને તેને સમજાવો

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:

```
int
a = 5,
b = 10;

if(a > 0 && b > 0)      //
if(a > 15 || b < 5)      //
if(!(a < 10))            //
```

મેમરી ટ્રીક

“અને અથવા નહીં”

પ્રશ્ન 3(b) [4 ગુણ]

for loop ને ઉદાહરણ સાથે સમજાવો.

જવાબ

For Loop માળખું:

ઘટક	હેતુ
Initialization	શરૂઆતી મૂલ્ય સેટ કરવું
Condition	ચલુ રાખવા માટે ટેસ્ટ
Update	Loop variable ને બદલવું

Syntax:

```
for(initialization; condition; update) \{
    statements;
\}
```

ઉદાહરણ:

```
// 1 5      print
for(int

i = 1; i {=} 5; i++) \{

    printf("\%d ", i);
\}
// 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) [7 ગુણ]

ગ્રાન્યુનિયન સંખ્યાઓ x અને y માંથી મહત્તમ શોધવા માટે પ્રોગ્રામ લખો.

જવાબ

```
\#include <stdio.h>

int main() \{
    int x, y, z, max;

    printf(" : ");
    scanf("\%d \ \%d \ \%d", \&x, \&y, \&z);

    max = x; //

    if(y {>} max) \{
        max = y;
    \}
```

```

if(z {} max) \{
    max = z;
\}

printf("      : \%d", max);

return 0;
\}

```

Algorithm ના પગલાં:

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

વૈકલ્પિક પદ્ધતિ:

```
max = (x {} y) ? ((x {} z) ? x : z) : ((y {} z) ? y : z);
```

મેમરી ટ્રીક

“માનો સરખાવો Update દર્શાવો”

પ્રશ્ન 3(a) OR [3 ગુણ]

conditional operator ને ઉદાહરણ સાથે સમજાવો.

જવાબ

Conditional Operator (Ternary):

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

ઉદાહરણો:

```
int  
  
a = 10,  
  
b = 20;  
  
int max = (a {} b) ? a : b;           // max = 20  
  
char grade = (marks {=} 60) ? {P} : {F};  
printf("Status: %s", (age {=} 18) ? "Adult" : "Minor");
```

સમાન if-else:

```
if(a {} b)  
    max = a;  
else  
    max = b;
```

ફાયદાઓ:

- સંક્ષિપ્ત: એક લીટીમાં expression
- કાર્યક્રમ: જડપી execution

મેમરી ટ્રીક

"પ્રશ્નચિહ્ન કોલન પરસંદગી"

પ્રશ્ન 3(b) OR [4 ગુણ]

while loop ને ઉદાહરણ સાથે સમજાવો.

જવાબ

While Loop:

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

ઉદાહરણ:

```
int i = 1;
while(i {=} 5) \{
    printf("\%d ", i);
    i++;
\}
// 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:

```
flowchart LR
    A[Initialize] --> B{Condition?}
    B --> C[Statements Execute]
    C --> D[Variable Update]
    D --> B
```

મેમરી ટ્રીક

"Initialize ચકાસો Execute Update"

પ્રશ્ન 3(c) OR [7 ગુણ]

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

જવાબ

```
#include <stdio.h>

int main() {
    int number;

    printf("Enter a number : ");
    scanf("\%d", &number);

    if(number \% 2 == 0) \{
        printf("\nEven number : %d", number);
    \}
    else \{
        printf("\nOdd number : %d", number);
    \}

    return 0;
}
```

તર્ક સમજૂતી:

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

વैकल्पिक पद्धतिओ:

```
//      2: Conditional operator
printf("\%d  \%s ", number, (number \% 2 == 0) ? " " : " ");

//      3: Bitwise AND
if(number \& 1)
    printf("  ");
else
    printf("  ");
```

Sample Output:

```
: 7
```

મેમરી ટ્રીક

“Modulus બે શૂન્ય સમ”

પ્રશ્ન 4(a) [3 ગુણ]

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

જવાબ

પગલાબદ્ધ મૂલ્યાંકન:

પગારું	Expression	ગણતરી	પરિણામ
1	$30/4*4$	$(30/4)*4 = 74$	28
2	$20\%6$	$20 \bmod 6$	2
3	$17/2$	Integer division	8
4	અંતિમ	$28 - 2 + 8$	34

Operator પ્રાધાન્યતા:

પ્રાધાન્યતા	Operators
ઉર્ચા	$*, /, \%$ (ડાબેથી જમણો)
નીચી	$+, -$ (ડાબેથી જમણો)

સંપૂર્ણ ગણતરી:

```
30/4*4 - 20\%6 + 17/2
= 7*4 - 2 + 8      //   division   modulus
= 28 - 2 + 8       //   Multiplication
= 26 + 8           //   +,-
= 34               //
```

મેમરી ટ્રીક

“ગુણા ભાગ પહેલાં બાદબાકી પછી”

પ્રશ્ન 4(b) [4 ગુણ]

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

જવાબ

```
\#include <stdio.h>

int main() {
    int numbers[5];
    int sum = 0;
    float average;

    printf("5           :{n}");
    for(int i = 0; i < 5; i++) {
        scanf("\%d", &numbers[i]);
        sum += numbers[i];
    }

    average = (float)sum / 5;

    printf("      = \%d{n}", sum);
    printf("      = \%.2f", average);

    return 0;
}
```

Algorithm:

પગલું	કાર્ય
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 કાર્યક્રમ

મેમરી ટ્રીક

“જાહેર Input ઉમેરો ગણો દર્શાવો”

પ્રશ્ન 4(c) [7 ગુણ]

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

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:

```
// Declaration
int *ptr;           // Integer pointer
float *fptr;         // Float pointer
char *cptr;          // Character pointer

// Initialization
int num = 10;
int *ptr = &num;    // num   address   initialize

//
int *ptr;
ptr = &num;          //      address assign
```

ઉદાહરણ પ્રોગ્રામ:

```
\#include <stdio.h>

int main() {
    int num = 25;
    int *ptr = &num;

    printf("num      : \%d\n", num);
    printf("num  address: \%p\n", &num);
    printf("ptr      : \%p\n", ptr);
    printf("ptr      pointed  : \%d\n", *ptr);

    return 0;
}
```

મુખ્ય Operators:

- & (Address-of): Variable જુદી address મેળવે છે
- * (Dereference): Address પરનું મૂલ્ય મેળવે છે

Memory Diagram:

```
num: [25] at address 1000
ptr: [1000] at address 2000
```

મેમરી ટ્રીક

“Address Star Dereference”

પ્રશ્ન 4(a) OR [3 ગુણ]

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

જવાબ

પગલાબદ્ધ મૂલ્યાંકન:

નંબર	Expression	ગણતરી	પરિણામ
1	$50/3$	Integer division	16
2	$16\%3$	$16 \bmod 3$	1
3	$5*7$	Multiplication	35
4	અંતિમ	$1 + 35$	36

સંપૂર્ણ ગણતરી:

```

50 / 3 % 3 + 5 * 7
= 16 % 3 + 35      // division multiplication
= 1 + 35           // Modulus operation
= 36                //

```

Operator પ્રાધાન્યતા લાગુ:

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

મેમરી ટ્રીક

“ભાગ Mod ગુણા ઉમેરો”

પ્રશ્ન 4(b) OR [4 ગુણ]

N પૂછાયોની array માં સૌથી મોટી સંખ્યા શોધવા માટેનો પ્રોગ્રામ લખો.

જવાબ

```

#include <stdio.h>

int main() {
    int n, i;
    int largest;

    printf("elements : ");
    scanf("\%d", &n);

    int arr[n];

    printf("\%d :{n}", n);
    for(i = 0; i < n; i++) {
        scanf("\%d", &arr[i]);
    }

    largest = arr[0]; // element largest

    for(i = 1; i < n; i++) {
        if(arr[i] > largest) {
            largest = arr[i];
        }
    }

    printf(" : \ \%d", largest);

    return 0;
}

```

Algorithm:

પગલું	કાર્ય
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

મेरી ટ્રીક

"Input માનો સરખાવો Update દર્શાવો"

પ્રશ્ન 4(c) OR [7 ગુણ]

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

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
ધણા variables ટાળવા	arr[100] બદલે a1, a2, ..., a100
કાર્યક્ષમ processing	Loop-based operations
Memory organization	Contiguous allocation

1-D Array Declaration:

```
datatype arrayname[size];  
  
//  
int marks[5];           // 5 integers   Array  
float prices[10];       // 10 floats    Array  
char name[20];          // 20 characters  Array
```

Array Initialization:

```
//      1: Declaration  
int numbers[5] = \{10, 20, 30, 40, 50\};  
  
//      2:      assignment  
int arr[3];  
arr[0] = 5;  
arr[1] = 15;  
arr[2] = 25;
```

ઉદાહરણ પ્રોગ્રામ:

```
\#include <stdio.h>  
  
int main() {  
    int marks[5] = \{85, 90, 78, 92, 88\};  
    int i, sum = 0;  
  
    printf("      :{n}");  
    for(i = 0; i \{ 5; i++) {  
        printf("  \%d: \%d{n}", i+1, marks[i]);  
        sum += marks[i];  
    }  
  
    printf("      : \%d", sum);  
    return 0;  
}
```

Memory Layout:

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

મેમરી ટ્રીક

“સમાન ડેટા સતત Index”

પ્રશ્ન 5(a) [3 ગુણ]

if ... else statement નું ઉદાહરણ આપો.

જવાબ

If-else ઉદાહરણ:

```
\#include <stdio.h>

int main() ^{
    int age;

    printf("           : ");
    scanf("\%d", &age);

    if(age {=} 18) ^{
        printf("           ");
    }
    else ^{
        printf("           ");
    }

    return 0;
}
```

માળખું:

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:
 Input: 16 Output:

મેમરી ટ્રીક

"જો સાચું નહીંતર ખોટું"

પ્રશ્ન 5(b) [4 ગુણ]

આપેલ character ની category અકાસવા માટેનો પ્રોગ્રામ લખો.

જવાબ

```
\#include <stdio.h>
#include <ctype.h>

int main() ^{
    char ch;

    printf(" character      : ");
    scanf("\%c", &ch);

    if(isdigit(ch)) ^{
        printf("{}\%c{      ", ch);
    }
    else if(isupper(ch)) {
```

```

        printf("{}\%c{      });
        \}
    else if(islower(ch)) \{
        printf("{}\%c{      ");
        \}
    else \{
        printf("{}\%c{      ");
        \}

    return 0;
\}

```

Character Categories:

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

વૈકલ્પિક પદ્ધતિ:

```

if(ch == '0' && ch == '9')
    printf(" ");
else if(ch == 'A' && ch == 'Z')
    printf(" ");
else if(ch == 'a' && ch == 'z')
    printf(" ");
else
    printf(" ");

```

મેમરી ટ્રીક

“અંક મોટા નાના વિશેષ”

પ્રશ્ન 5(c) [7 ગુણ]

Structure શું છે? તેની syntax ખોઝ્ય ઉદાહરણ સાથે સમજાવો

જવાબ

Structure વ્યાખ્યા:

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

Syntax:

```
struct structure\_name \{
    datatype member1;
    datatype member2;
    ...
\;}
```

ટ્રાન્સ્ફરણ - Student Structure:

```
\#include <stdio.h>

struct Student \{
    int roll\_no;
    char name[50];
    float marks;
    char grade;
\;}

int main() \{
    struct Student s1;

    // Data input
    printf(" : ");
    scanf("\%d", \&s1.roll\_no);

    printf(" : ");
    scanf("\%s", s1.name);

    printf(" : ");
    scanf("\%f", \&s1.marks);

    printf(" : ");
    scanf(" \%\c", \&s1.grade);

    // Data display
    printf("{n} :{n}");
    printf(" : \%d{n}", s1.roll\_no);
    printf(" : \%s{n}", s1.name);
    printf(" : \% .2f{n}", s1.marks);
    printf(" : \%c{n}", s1.grade);

    return 0;
\}
```

Structure લક્ષણો:

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

Structure Initialization:

```
struct Student s1 = \{101, "John", 85.5, {A}\;}
```

Memory Layout:

```
s1: [roll_no] [name...] [marks] [grade]
    4 bytes 50 bytes 4 bytes 1 byte
```