**Second Year Computer Science**  
**Option – I (Programming Using C)**  
**Dada Education Official PK – Preparation Paper Answer Key**

**SECTION B – Short Questions**

**Q2.**

**(i)**

* **Library Functions**: Built-in functions in C.
  + Examples: printf(), scanf(), strcpy(), sqrt()
* **Address Operator**: & is used to get the address of a variable.
  + Example: int a = 5; printf("%p", &a);

**(ii)**

* **Steps to Create Table in MS Access**:
  1. Open MS Access
  2. Create a new database
  3. Select “Table Design”
  4. Define fields and data types
  5. Set primary key
  6. Save the table
* **Message Sequence**: Order in which messages are processed.
  1. Examples: Welcome message, error message

**(iii)**

* Reserved words **cannot** be used as variable names in C because they have predefined meanings.
  + Examples: int, float, return

**(iv)**

* **C Preprocessor Directives**: Instructions processed before compilation.
  + #define: Macro definition
  + #include: Includes standard/header files
* **IDE Shortcut Keys**:
  + Ctrl + S (Save), Ctrl + Z (Undo), Ctrl + C (Copy)

**(v)**

* **Factorial Program**:

int factorial(int n) {

if(n == 0) return 1;

return n \* factorial(n - 1);

}

* **Logical Operators**: &&, ||, !
  + Example: if(a > 0 && b > 0)
* **Pointer in C**: Stores the address of another variable.
  + Declaration: int \*p;
  + Initialization: p = &a;

**(vi)**

* **scanf()**: Used to take input.
  + Syntax: scanf("%d", &num);
* **Data Redundancy**: Repetition of data in a database.
* **Reserved Words in C**: int, float, char, return, if, else, etc.

**(vii)**

* **Built-in Functions with Examples**:
  + abs(x), pow(x, y), sqrt(x), fopen(), scanf(), printf()

**(viii)**

* **Differences**:
  + **Primary Key vs Foreign Key**: Unique vs reference key
  + **Source vs Object Code**: Human-readable vs machine-readable
  + **Increment/Decrement**: ++, --
  + **Internal/External Variables**: Local vs global scope

**(ix)**

* **Program to Generate Output**:

for(int i=1; i<=3; i++)

printf("%d\n", i\*i);

* Output:

1

4

9

**(x)**

* **Definitions**:
  + **DBMS**: Software for managing databases
  + **Data Dictionary**: Stores metadata
  + **Function Prototype**: Declares function signature
  + **Algorithm**: Sequence of steps
  + **Notation**: Algorithm writing format
  + **Table**: Structure of data
  + **Query**: Data retrieval request
  + **Break/Continue**: Loop control statements

**(xi)**

* **Function to Return Maximum**:

int max(int a, int b) {

return (a > b) ? a : b;

}

**(xii)**

* **C Program Equivalents**:

A = ((x\*y)+(x-y))/(x+y);

A = 3.14\*r\*r;

A = (3\*a)/7;

C = sqrt((b\*b) - (4\*a\*c))/(2\*a);

OR C = v + b + d;

D = (x\*y)+(z\*x) - 3\*xyz;

**(xiii)**

* **Program to Calculate Area of Rectangle**:

int length = 5, width = 3;

int area = length \* width;

* **Scope of Identifiers**:
  + Local: Within function/block
  + Global: Outside all functions

**(xiv)**

* **Strings**: Array of characters ending with \0
* **If-Else Structure**:

if(condition) {

// statements

} else {

// statements

}

* **Basic C Program**:

#include<stdio.h>

int main() {

printf("Hello World");

return 0;

}

**(xv)**

* **Loop Program to Print Patterns**:

for(int i=1; i<=5; i++) {

for(int j=1; j<=i; j++)

printf("\*");

printf("\n");

}

**SECTION C – Long Questions**

**Q3.**

* **Function**: A block of reusable code.
* **Advantages**: Reusability, modularity, clarity
* **Function Prototype**: Declares return type and parameters
  + Example: int sum(int, int);
* **Switch & Break Example**:

switch(choice) {

case 1: break;

default: break;

}

**Q4.**

* **Database Models**:
  + Hierarchical
  + Network
  + Relational
* **Data Types in C**:
  + int, float, char, double, void

**Q5.**

* **Control Structure Types**:
  + Sequential
  + Selection (if, switch)
  + Iteration (loops)
* **Iteration Example**:

for(int i=0; i<5; i++) {

printf("%d ", i);

}

**Q6.**

* **Decision Structures in C**:
  + if, if-else, nested if, switch
* **Loops in C**:
  + for, while, do-while
* **Nested Loop Example**:

for(int i=1;i<=3;i++) {

for(int j=1;j<=3;j++)

printf("\* ");

printf("\n");

}

**Q7.**

* **If-Else Statement**:
  + Used for conditional execution
* **Difference from Switch**:
  + if: Handles ranges and logical conditions
  + switch: Handles discrete values
* **Database Relationship Types**:
  + One-to-One
  + One-to-Many
  + Many-to-Many
* **Example**: A student enrolled in multiple courses

**End of Document**