

CS1121
Shyed Shahrear MCE 07905536
Ans. to the q. no-1

①

#include <stdio.h> // Includes the standard input/output library for functions like printf

int main(void) { // This line defines main function, entry point, returns integer value and takes no argument.

int ctr; // declares an integer variable ctr, without initializing it.

for(ctr=65; ctr<91; ctr++) // This line starts a for loop, ctr is initialized at 65, the loop runs as long as ctr is less than 91 and at each iteration of the loop ctr is incremented by 1.

printf("%c", ctr); // prints a character represented by the value stored in ctr; %c is format specifier to print char value of int ctr.

Shyed Shahriar MC E 07905536 ②

return 0; // the main function is returning an integer value of 0 to the operating system, successful execution of program

The output of this program will be capital english A to Z, ASCII 65 to 90

Re changed code is (allowing warning)

```
# int main (void) {  
    for (int ctr=65; ctr<91; ctr++)  
        printf ("\t%c", ctr);  
    return 0;  
}
```

Shyed Shahriar MCE 07905536

③

Ans. to the q.no-2

Yes, the right side code will compile.

Corrected for a space in input line and rewritten for more readability

```
#include <stdio.h>
int x,y;
```

```
int main() {
```

```
{ printf("\n Enter two numbers\n");
```

```
scanf ("%d %d", &x, &y);
```

```
printf ("\n \n %d is bigger",
```

```
(x > y)? x:y);
```

```
return 0; }
```

Shyed Shahriar MCE07905536

(4)

Ans. to q. no-3

Compile time errors: Occurs during translation the code into machine code by compiler, program is not built. typically caused by syntax errors. Detected by compiler, exe is not created.

Runtime error: Run time error or execution error occurs when the program is running or executed resulting in false results. This can not be detected by compiler. Logical errors.

The provided code will not return a compile time error, but it has a logical error. The array declaration is not correct, will cause runtime error. Declaring an array of size 10737 but accessing

Shyed Shahriar MCE07905536 ⑤

as if it was 3×10 array, it will have unexpected behaviour.
useful code will be

```
#include <stdio.h>
int main(void)
{
    int array[3][10];
    int x, y;
    int ctr=0;
    for (x=0; x<3; x++)
        for (y=0; y<10; y++)
            array[x][y] = ctr;
            ctr++;
    printf("%d ", array[x][y]);
    {
        printf("\n");
    }
    return 0;
}
```

This will print 0 to 29 in 3 lines.

Shyed Shahriar MCE 07905536

⑥

Ans. to q. no-4'

Rewriting the if-else if statement
task with case is

int num;
printf("Enter choice"); scanf("%d", &num);
switch (num)

case (test expression 1) {

// Statement(s)

break;

case (test expression 2) {

// Statement(s)

break;

default {

// Statement(s)

}

Ans. to q. NO-5

Pointers are fundamental concepts and tools in C programming giving several advantages:

- ① Memory management
- ② Direct Memory access
- ③ Efficiency
- ④ Use less memory to copy variables
- ⑤ Faster
- ⑥ Many functionality - DSA, Algorithm, arrays.
- ⑦ Dynamic memory allocation.
- ⑧ Hardware control capability

// C program to add two numbers with pointers;

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

```
int main(void){
```

```
    int num1, num2, sum, *ptr1, *ptr2;
```

```
    printf("Enter 2 numbers : ");
```

```
    scanf("%d,%d", &num1, &num2);
```

```
    ptr1 = &num1;
```

```
    ptr2 = &num2;
```

```
    sum = *ptr1 + *ptr2;
```

```
    printf("sum of %d and %d is %d \n", *ptr1, *ptr2, sum);
```

```
    return 0;
```

Ans. to q. no - 6

Recursion is a programming technique, in which a function calls itself to solve problems and each subproblem is solved using the same function.

/* C program to calculate factorial using recursion

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

```
unsigned long long factorial (int n) {
```

```
if (n==0 || n==1) {
```

```
    return 1; }
```

```
else { return n*factorial (n-1); }
```

```
int main() {
```

```
int num;
```

```
printf ("Enter a positive integer: ");
```

```
scanf ("%d", &num);
```

(P.T.O)

Shyed MCE 07905536

⑨

```
if (num < 0) {  
    printf ("incorrect input");  
}  
else {  
    unsigned long long result = factorial (num);  
    printf ("Factorial of %d is %llu\n",  
           num, result);  
}  
return 0;  
}
```

(P.T.O)

Ans. to q. no-7

```
#include<stdio.h>
#include <string.h>

#define Max Student 100
#define Max Roll length 20
#define Max Name length 50
#define Max subjects 5
#define Max Sub Name Length 10

typedef struct { char roll [Max Roll length];
                 int marks [Max subjects];
                 } Student Marks;

typedef struct { char roll [Max Roll length];
                 char name [Max Name Length];
                 } Student;

float calculateGPA (int marks) {
    if (marks >= 90) return 4.0;
    else if (marks >= 80) return 3.5;
    else if (marks >= 70) return 3.0;
    else if (marks >= 60) return 2.5
}
```

(P.T.O)

```
else if (marks == 50) return 0;  
else return 0.0;
```

{

```
char calculateLetterGrade(int marks){  
    if (marks >= 90) return 'A';  
    else if (marks >= 80) return 'B';  
    else if (marks >= 70) return 'C';  
    else if (marks >= 60) return 'D';  
    else return if (marks >= 50) return 'P';  
    else return 'F'; }
```

```
int main(){
```

```
File *InfoF = fopen("StdInfo.txt", "r");  
File *MarkF = fopen("StdMark.txt", "r");  
File *O.P = fopen("Stdgrades.txt", "w");  
if (InfoF == Null || MarkF == Null ||  
    O.P == Null){ perror("Error");  
    return 1; }
```

```
int roll, marks1, marks2, marks3;  
char name[50];  
  
while(fscanf(stdin, "%d %s", &roll, name) != EOF  
    &  
    fscanf(stdin, "%d %d %d", &marks1, &marks2, &marks3) != EOF){  
    float gpa1 = calculateGPA(marks1);  
    float gpa2 = // similar code  
    // as above  
  
    char float grade = calculateLetterGrade(marks1);  
    // similar code  
    // as above  
  
    float finalGPA = (gpa1 + gpa2 + gpa3)/3.0;  
    char avgGrade = calculateLetterGrade((marks1 +  
        marks2 + marks3)/3);
```

fptrints(~~output~~, OF, "RollId/n, Name: Y.SN,

- Letter grade l.: Y.C, GPA: Y.2f \n, Total GPA;
- - - - - Q // Similar code for all 3 subjects

Y. 2f \n, Average grade: Y.C \n \n,"
roll, name, grade1, gpa1, grade2,gpa2
----- total(GPA, avgGPA);

fclose (InfoF);

fclose (marktF);

fclose (OF);

printf ("Written to stdgrades.txt");

return 0;

}