

## H.W.P

EXERCISE (CH # 2)

②.1 Fill in the blanks;

a) main      b) { , }      c) semicolon ( ; )

d) printf()      e) new : line      f) scanf()

g) %d      h) destructive      i) nondestructive

j) if

\_\_\_\_\_ x \_\_\_\_\_ x \_\_\_\_\_ x

②.2 True / False - If False, explain why;

a) False (because if we want to print output at the beginning of new line we use escape sequence \n)

b) False (because comments are not executed it only help to make our code more readable)

c) True

d) True

e) True

⑤ False (because C language is case-sensitive and the meanings of 'a' and 'A' are different in C language).

⑥ False (The definition of function must appear after the left-brace '{' of the body of function).

⑦ False (Arguments in a printf function should not be preceded by ampersand - Arguments following the format control string in scanf function should be preceded by ampersand).

i) True

j) False (Multiplication \*, Division / and Remainder % have higher precedence than Addition + and Subtraction - operators).

k) False (Some systems may distinguish between identifiers longer than 31 characters).

l) False (T. print these as multiple

lines output we can use escape sequence '\n' in a single printf function).

Q. Write a single C statement;

a. int d, this Variable, 976354, number;

b. printf("Entered an integer: ");

c. scanf ("%d", &a);

d. if (number != 7) {  
    printf("The variable number is not equal to 7.");  
}

e. printf("This is a C program.\n");

f. printf("This is a C\nprogram.\n");

g. printf("This\nis\nmain\nin\nProgram");

h. printf("This\\t is \\t a \\t C \\t program.");

i. printf("This\\t is \\t a \\t C \\t program.");

(2.4) Write a statement (or comment);

① /\* Program will calculate the product  
of three integers. \*/

② int x, y, z, result;

③ printf("Enter three integers");

④ scanf("%d%d%d", &x, &y, &z);

⑤ result = x \* y \* z;

⑥ printf("The product is %d", result);

(2.5) Using (2.4) statements, write a  
proper program;

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

```
int main()
```

```
    int x, y, z, result;
```

```
    printf("Enter three integers");
```

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

```
    result = x * y * z;
```

```
    printf("The product is %d", result);
```

```
}
```

(2.6) Identify and correct the errors;

① printf("The value is %d\n", number);

② scanf("%d%d", &number1, &number2);

③ if (c > 7) {

printf("C is equal to or less  
than 7\n");

```
}
```

④ if (c >= 7) {

printf("C is equal to or less than 7\n");

```
}
```

(2.7) Identify and correct the errors

① scanf("%d", &value);

② printf("The product of %d and %d  
is %d\n", x, y, x \* y);

③ sumOfNumbers = firstNumber + secondNumber;

④ if (number >= largest) {

largest == number;

```
}
```

② /\* Program to determine the largest of three integers \*/  
③     scanf("%d", &nInteger);

④ printf("Reminder of %d divided by %d is %d\n", n, y, n % y);

⑤ if (x == y) {  
    printf("x is equal to y.\n", x, y);  
}

⑥ printf("The sum is %d\n", x+y);

⑦ printf("The value you entered is: %d\n", value);

⑧ Fill in the blanks;

⑨ Comments : ⑩ printf()

⑪ if statement ⑫ arithmetic

⑬ scanf()

⑭ False (same reason as ⑮ part ↑).

⑮ False (because variable name can not be start with number).

2.9 Write a single C statement:

① printf("Enter two numbers.");

② a = b \* c;

③ /\* Program to perform a simple payroll calculation \*/  
④ scanf("%d %d %d", &a, &b, &c);

2.10 True / False - A) False, explain why;

① False (compiler first evaluate the expressions which have higher precedence, if operators of same precedence occurs then associativity rule is performed which is left to right) -

② True

③ True

④ False (same reason as ⑤ part ↑).

2.11 Fill in the blanks

① Division / and Remainder %

② innermost parenthesis

③ Variable

2.12 Write outputs of the following statements;

Assume  $x=2$  and  $y=3$

④ 2      ⑤ 4

⑥  $x=$     ⑦  $x=2$

⑧  $5 = 5$     ⑨ Nothing

⑩ Nothing    ⑪ Nothing

⑫ Blank line

2.13 MCQ's

① True (✓)

\_\_\_\_\_ x \_\_\_\_\_ x \_\_\_\_\_ x \_\_\_\_\_

2.14 MCQ's

② True (✓)

\_\_\_\_\_ x \_\_\_\_\_ x \_\_\_\_\_ x \_\_\_\_\_

2.15 What is the value of 'x'?

③ 15      ④ 4

⑤ 216

\_\_\_\_\_ x \_\_\_\_\_ x \_\_\_\_\_ x \_\_\_\_\_

2.16 Program

```
#include<stdio.h>
int main() {
    print int a, b;
    printf("Enter the value of 'a' and 'b':");
    scanf("%d%d", &a, &b);
    printf("Sum is: %d\nProduct is: %d\n"
           "Difference is: %d\nQuotient is: %d\n"
           "Remainder is: %d\n", a+b, a*b, a-b, a/b, a%b);
```

return 0;

}

x

### (2.17) Program

```
#include<stdio.h>
int main() {
    int a, b;
    printf("Enter the value of 'a' and
'b':");
    scanf("%d %d", &a, &b);
    if (a > b) {
        printf("%d is largest\n", a);
    } else if (b > a) {
        printf("%d is largest\n", b);
    } else {
        printf("These numbers are equal\n");
    }
}
return 0;
```

### (2.18) Program

2.19

## Program

printf("i.d is largest\n", c);

```
#include<stdio.h>
int main() {
    int a, b, c, sum, average, product;
    printf("Enter the value of 'a', 'b' and
    'c': ");
    scanf("%d%d%d", &a, &b, &c);
    sum = a+b+c;
    average = (a+b+c)/3;
    product = a*b*c;
    printf("Sum is %d\nProduct is %d\n",
    "Average is = %d\n", sum, product, average);
    if (a>b) {
        if (a>c) {
            printf("i.d is largest\n", a);
        } else {
            printf("i.d is largest\n", c);
        }
    } else {
        if (b>c) {
            printf("i.d is largest\n", b);
        } else {
            printf("i.d is largest\n", c);
        }
    }
    return 0;
}
```

2.20

## Program

#include <stdio.h>

```
int main() {
    float r, diameter, circumference,
        areaOfCircle, pi = 3.14159;
    printf("Enter the radius of
        circle : ");
    scanf("%f", &r);
```

$$\text{diameter} = \gamma * \gamma;$$

$$\text{circumference} = 2 * \pi * r;$$

$$\text{areaOfCircle} = \pi * (\text{radius} * \text{radius});$$

```
printf("Diameter of Circle is = %f\n",
```

Circumference of Circle is =  $\pi r$

Area of Circle is =  $\pi r^2$

diameter, circumference area of circle);

return 0;

2

9:21

## Program

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

return 0;

Output is



2.22 Write output;

\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*

x x x

2.23 Program

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

```
int main() {
```

```
    int a, b, c, d, new1L, new2L, new3L,  
        new1S, new2S, new3S;
```

```
    printf("Enter the value of a : ");  
    scanf("%d", &a);
```

```
    printf("Enter the value of b : ");  
    scanf("%d", &b);
```

```
    printf("Enter the value of c : ");  
    scanf("%d", &c);
```

```
    printf("Enter the value of d : ");  
    scanf("%d", &d);
```

$$\begin{aligned} \text{new1L} &= (a > b) * a + (b > a) * b; \\ \text{new2L} &= (\text{new1L} > c) * \text{new1L} + (c > \text{new1L}) * c; \\ \text{new3L} &= (\text{new2L} > d) * \text{new2L} + (d > \text{new2L}) * d; \end{aligned}$$

```
printf("Largest value is = %d\n", new3L);
```

$$\text{new1S} = (a < b) * a + (b < a) * b;$$

$$\begin{aligned} \text{new2S} &= (\text{new1S} < c) * \text{new1S} + (c < \text{new1S}) * c; \\ \text{new3S} &= (\text{new2S} < d) * \text{new2S} + (d < \text{new2S}) * d; \end{aligned}$$

```
printf("Smallest value is = %d\n", new3S);
```

```
return 0;
```

2.24

Program

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

```
int main() {
```

```
    int a;
```

```
    printf("Enter the value of a : ");  
    scanf("%d", &a);
```

```
    if (a % 2 == 0) {  
        printf("Even\n");  
    }
```

```
if (a % 2 != 0) {  
    printf("Odd\n");  
}
```

2.7.6. Log xam

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

int main() {  
 int a, b;  
 printf("Enter the value of a : ");  
 scanf("%d", &a);  
 printf("Enter the value of b : ");  
 scanf("%d", &b);  
}

return 0;

Output

D D D D D D D D D D

ପ୍ରକାଶକ

2.27

#include <stdio.h>

```
int main () {
```

```
    printf('*.*.*.*.*.*\n');
```

```
*.*.*.*.*.*\n
```

```
*.*.*.*.*.*\n
```

```
*.*.*.*.*.*\n
```

```
*.*.*.*.*.*\n
```

```
return 0;
```

}

2.28 #include <stdio.h>

int main() {  
 printf("Character's set values of C (ASCII)-  
 Integer values of the following character  
 are given below:\n");

```
printf("A = %d\n", 'A');
```

```
printf("B = %d\n", 'B');
```

```
printf("C = %d\n", 'C');
```

```
printf("a = %d\n", 'a');
```

```
printf("b = %d\n", 'b');
```

```
printf("c = %d\n", 'c');
```

```
printf("Q = %d\n", 'Q');
```

```
printf("L = %d\n", 'L');
```

```
printf("2 = %d\n", '2');
```

```
printf("S = %d\n", 'S');
```

```
printf("I = %d\n", 'I');
```

```
printf("T = %d\n", 'T');
```

cause the program stops execution so we  
know that there is an error, but  
in the other hand when non-fatal  
error the program continue to execute  
and outputs wrong results. Therefore  
in some cases we could not know  
there is a non-fatal error.

2.29 #include <stdio.h>

int main() {  
 printf("Character's set values of C (ASCII)-  
 Integer values of the following character  
 are given below:\n");  
 printf("A = %d\n", 'A');  
 printf("B = %d\n", 'B');  
 printf("C = %d\n", 'C');  
 printf("a = %d\n", 'a');  
 printf("b = %d\n", 'b');  
 printf("c = %d\n", 'c');  
 printf("Q = %d\n", 'Q');  
 printf("L = %d\n", 'L');  
 printf("2 = %d\n", '2');  
 printf("S = %d\n", 'S');  
 printf("I = %d\n", 'I');  
 printf("T = %d\n", 'T');

allows a program to continue  
executing, but causes it to  
produce incorrect results.  
→ We might get fatal error

```

    printf("Blank-Space = %d\n", ' ');
    return 0;
}

```

$A=65$ ,  $B=66$ ,  $C=67$ ,  $a=97$ ,  
 $b=98$ ,  $c=99$ ,  $0=48$ ,  $1=49$ ,  
 $2=50$ ,  $$=36$ ,  $*=42$ ,  
 $+ = 43$ ,  $/ = 47$ , Blank-Space = 32

### Character set of Computer

"The set of characters a computer uses together with the corresponding integer representations for these characters is called that computer's character set".

(2-30)

### Program

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

```

int main() {
    int number;
    printf("Enter five-digit number : ");
    scanf("%d", &number);
}

```

```

int quot1 = number / 10;
int rem1 = number % 10;
int quot2 = quot1 / 10;
int rem2 = quot1 % 10;
int quot3 = quot2 / 10;
int rem3 = quot2 % 10;
int quot4 = quot3 / 10;
int rem4 = quot3 % 10;

```

```

printf("%d...%d...%d...%d\n",
       quot4, rem4, rem3, rem2, rem1);

```

```

    return 0;
}

```