Institute of Computer Technology

B. Tech. Computer Science and Engineering

Sub: ESFP – I

Assignment-1

Date of Submission:10-11-2022

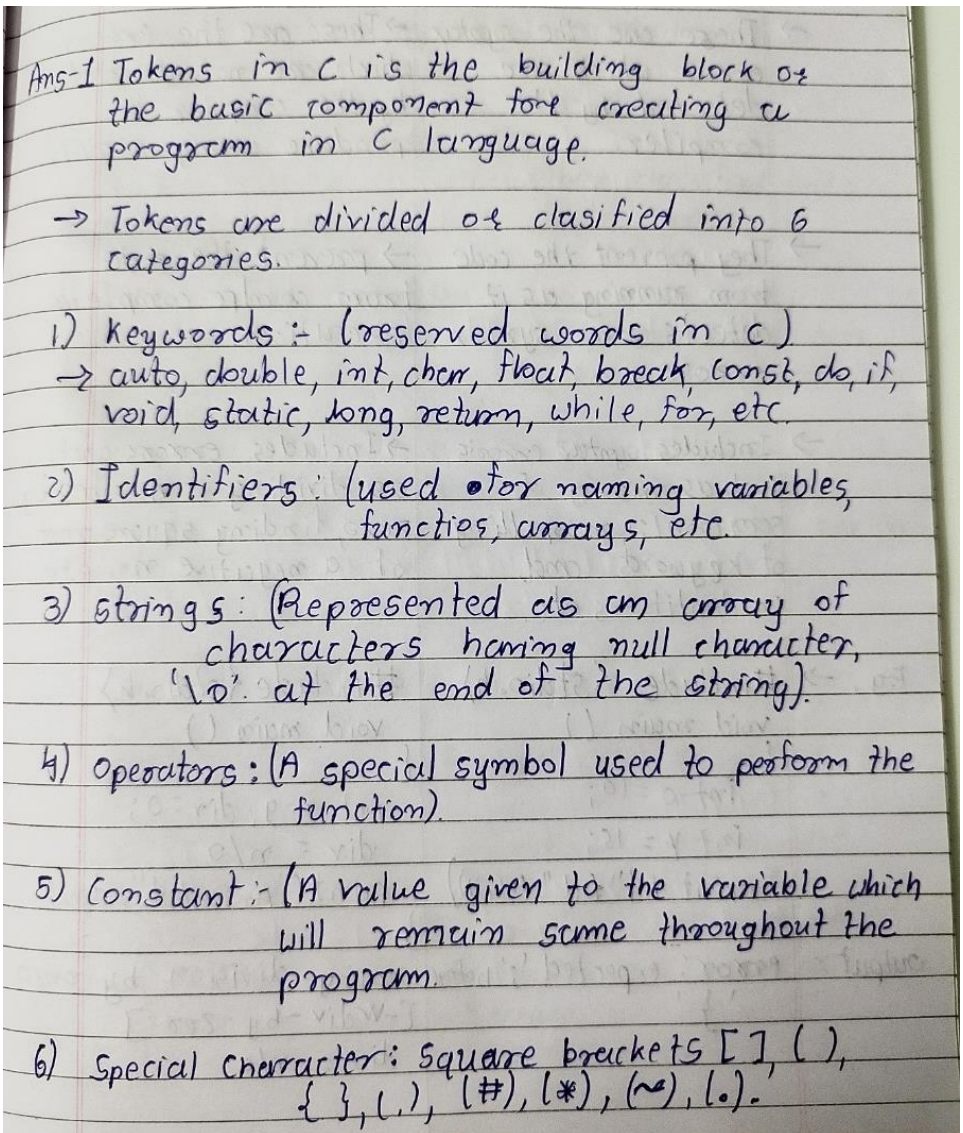
Name: Patel Ayush Padminkumar

Branch: Cyber security

Enrolment no: CS 32 batch-13

Answer the following questions:

1. What are Tokens? Explain various kinds of tokens in c programming language?



Ans-1 Tokens in C is the building block or the basic component for creating a program in C language.

→ Tokens are divided or clasified into 6 categories.

1) Keywords :- (reserved words in C)
→ auto, double, int, char, float, break, const, do, if, void, static, long, return, while, for, etc.

2) Identifiers : (used for naming variables, functios, arrays, etc.

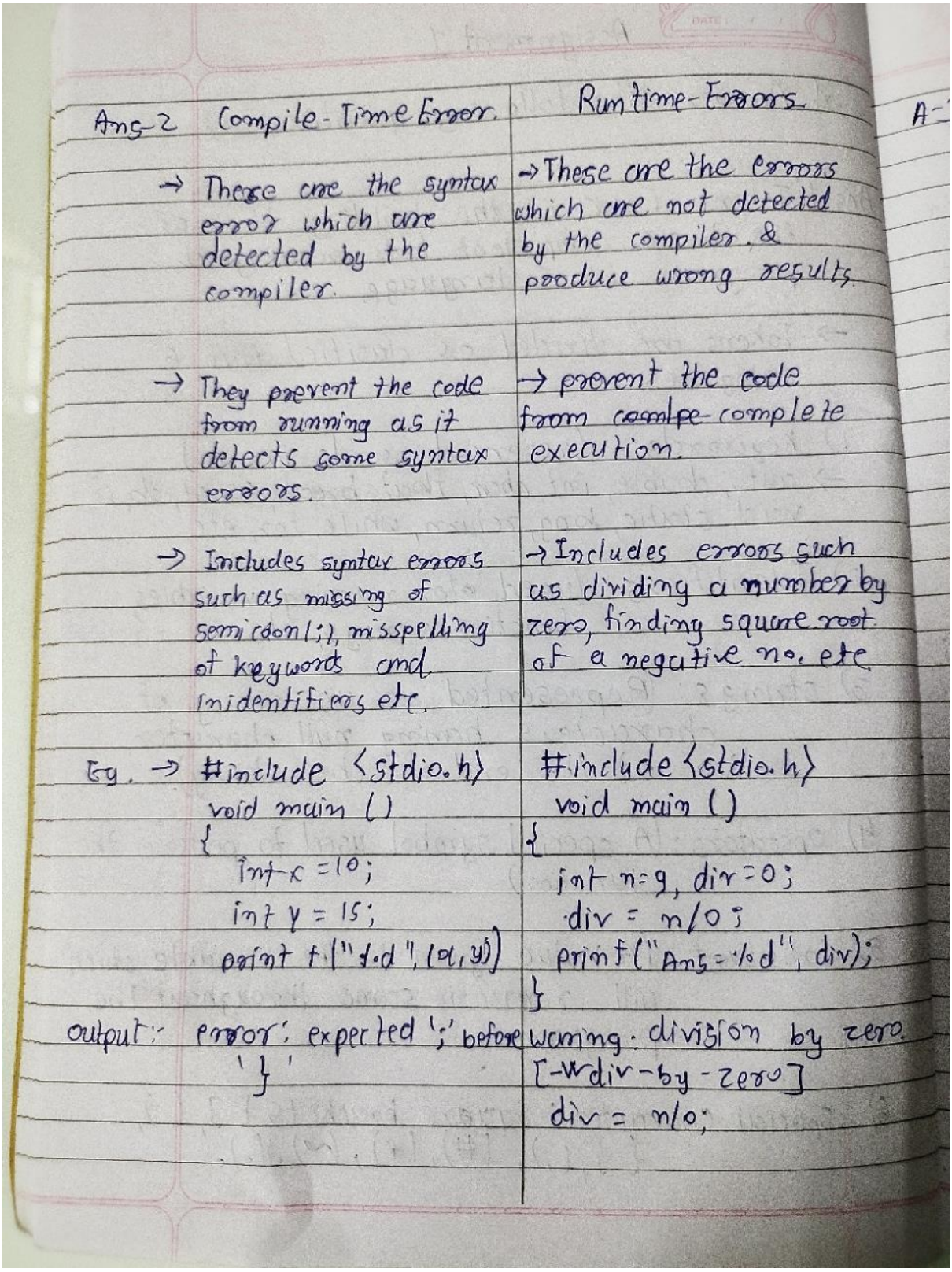
3) Strings : (Represented as an array of characters having null character, '\0' at the end of the string).

4) Operators : (A special symbol used to perform the function).

5) Constant :- (A value given to the variable which will remain same throughout the program.

6) Special Character : Square brackets [ ], ( ), { }, (,), (#), (*), (~), (.).

2. Explain the differences between compile time and run time errors with the help examples?



Ans-2

Compile-Time Error	Run time-Errors
→ These are the syntax error which are detected by the compiler.	→ These are the errors which are not detected by the compiler & produce wrong results.
→ They prevent the code from running as it detects some syntax errors.	→ prevent the code from ~~compile~~ complete execution.
→ Includes syntax errors such as missing of semicolon (;), misspelling of keywords and identifiers etc.	→ Includes errors such as dividing a number by zero, finding square root of a negative no. etc.

Eg. → `#include <stdio.h>`
`void main ()`
`{`
`int x = 10;`
`int y = 15;`
`printf("%d", (x, y))`
	`#include <stdio.h>`
	`void main ()`
	`{`
	`int n=9, div=0;`
	`div = n/0;`
	`printf("Ans = %d", div);`
	`}`

output: error; expected ';' before '}'	warning: division by zero [-Wdiv-by-zero] `div = n/0;`

3. Explain different types of data types in C with a suitable program example.

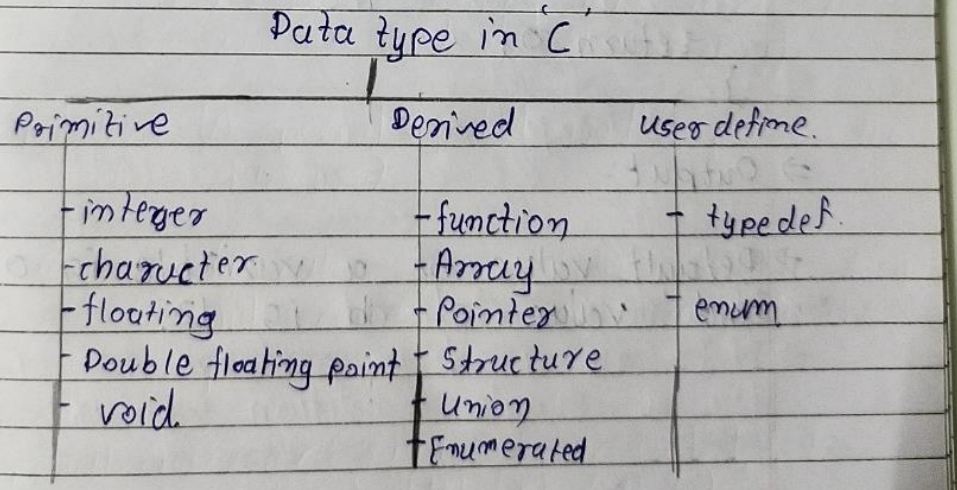
A-3 Data types in c:-

→ Data types in c can be classified in three groups.

1) Primitive Data type:-
→ Arithmetic types can be further classified into integer and floating data types.

2) User define data type:-
→ Mainly used assign names to integral constans, which make a program easy to read and maintain

3) Derived type:-
→ The data type that are derived from the primitive or built-in datatype are refered as derived data type.



4. Which variables below are syntactically correct?

✓ Num_var

✓ 1a

✓ float

✓ sum’s

✓ var1 var2

✓ v2

✓ r.no

✓ i

Ans:

- Num_var
- var1 var2
- v2
- i

5. What will happen if a variable is declared as constant? Give a program example for constant variable declaration.

A-5) Variables can be declared as constant using the "const" keyword before the data-type of the variable or the # define preprocessor directive. The constant variable can be ~~initi~~ initialized once only. The default type of constant variables are zero.

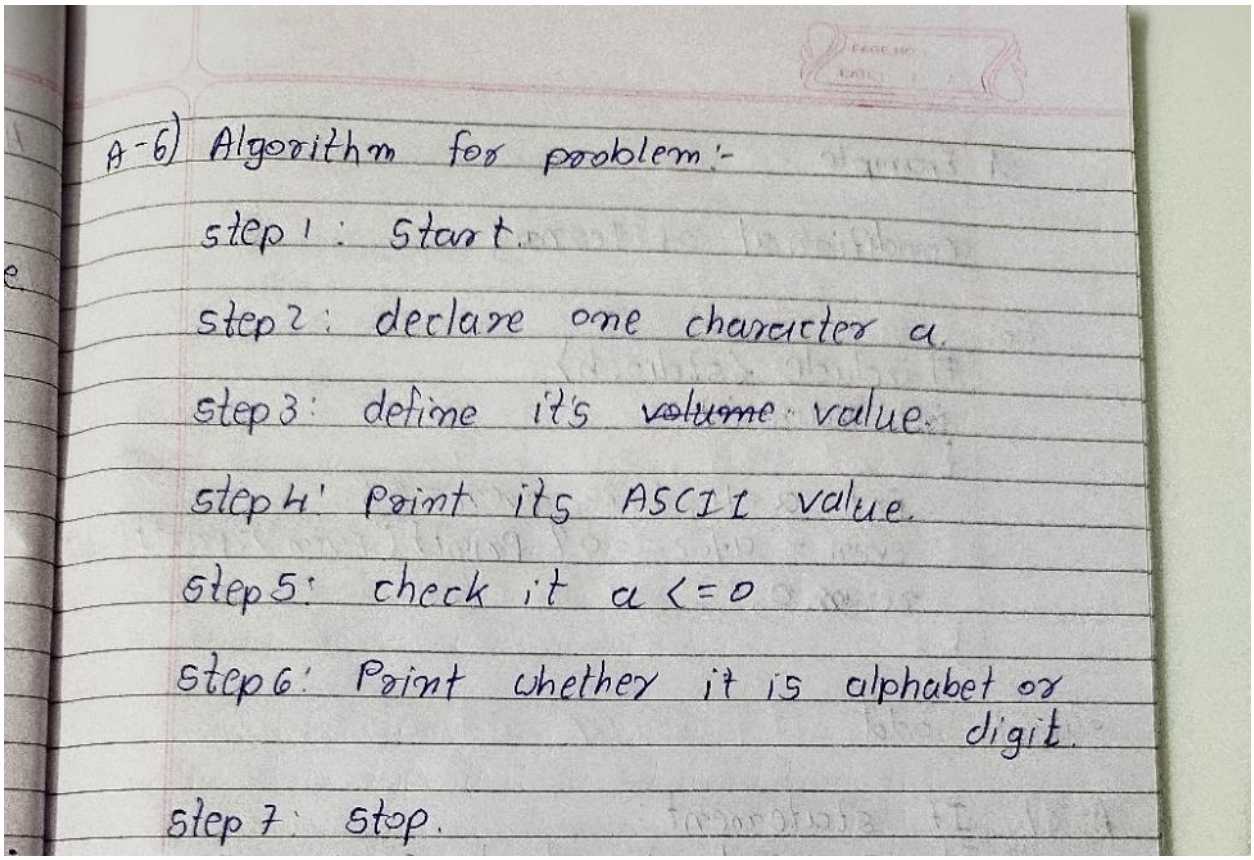
⇒ Example:-

```
#include <stdio.h>
 int main ()
{
 Const int a;
    const int b = 12;

 printf("default value of a variable=%d", a);
 printf ("default value of b =%d", b);

 return 0;
}
```

⇒ Output:-

→ Default value of a variable is 0 and value of b is 12.

6. Write an algorithm that allows for the input of a character value, prints its ascii code and displays whether it is alphabet or digit?



A-6) Algorithm for problem:-

step 1: Start.

step 2: declare one character a.

step 3: define it's ~~volume~~ value.

step 4: Print its ASCII value.

step 5: check it a <= 0

step 6: Print whether it is alphabet or digit.

step 7: stop.

7. Explain different types of operators in C with suitable program examples?

A-7 Types of operators :-

1) Arithmetic operator (+, -, *, /, %)
2) Relation Operator (<, <=, >, >=, ==, !=)
3) Logical operator (&&, ||, !)

4) Assignment operator (=)
5) Increment and decrement operator (++, --)
6) Conditional operators (condition? True or False.
7) Bitwise operators (&, |, ^, <<, >>, ~)
8) Special operator.
(comma operator) (Sizeof operator.

A Example :-

Conditional or Ternary

Ex:

```
#include <stdio.h>
int main ()
{
   int a=11, b=20, even = 0;
   even = a%2==0? Printf("Even"): printf("odd");
   return 0;
}
```

output:- odd

8. Explain the concept if, if – else, nested if – else and else if ladder condition in c with the help of program example.

A-8) If statement:
→ It is used to check some given condition and perform some operations depending upon the correction of that condition syntax:

```
if (expression) {
    // code to be executed
}
```

→ If-else statement:
→ It is used to perform two operations for a single condition syntax:

```
if (expression) {
    // code to be executed if condintio is then
}
else {
    // code executed if condition is false
}
```

⇒ Else-if ladder:
→ It is used as an extension to the if-else statement.

Syntax:-

⇒ condition-1

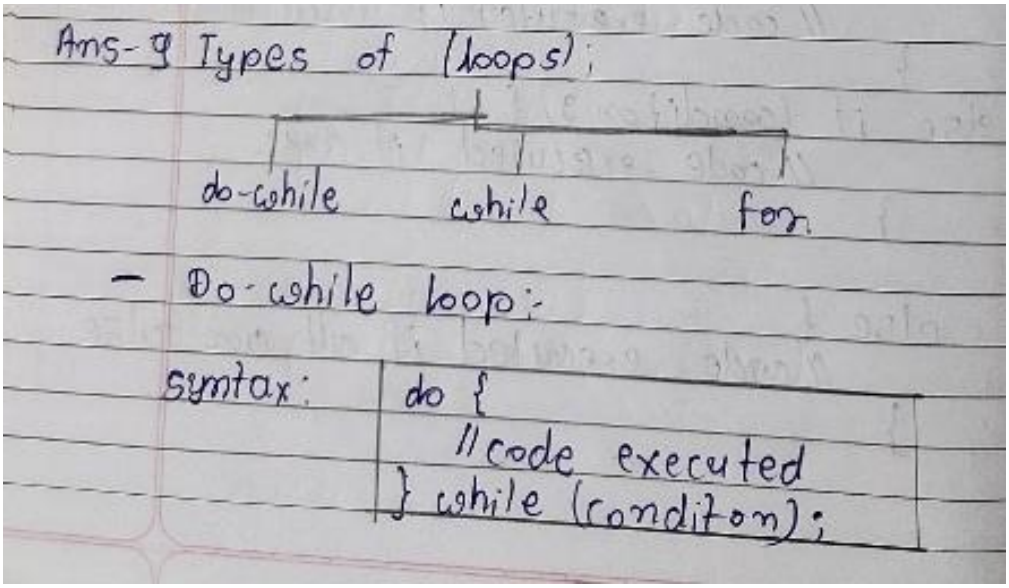
```
if {
    // code performed if true.
}
else if (condition 2){
    // code executed if true.
}
else if (conditon 3) {
    // code executed if true.
}
...
else {
    // code executed if all are false.
}
```

⇒ Example (if-else)

```
#include <stdio.h>
int main()
{
  int age;
  printf("Enter your age: ");
  scanf("%d", age);

if (age >= 18){
    printf("you are eligible to vote...");
}
else {
    printf("you are not eligible to vote");
}

  return 0;
}
```

9. What are the classifications of loops? Explain with the help of syntax, flowchart, and suitable program examples?

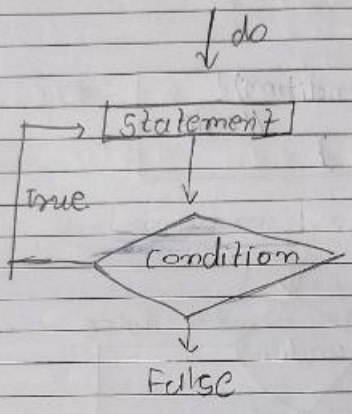
Ans- 9 Types of (loops):



- Do-while loop:

syntax:	do {
	// code executed
	} while (conditon);

⇒ Flowchart:



Example:

```
#include <stdio.h>
  int main ()
{
  int i=1;
  do {
    printf("%d", i);
  i++;
} while (i<=100);

  return 0;
}
```

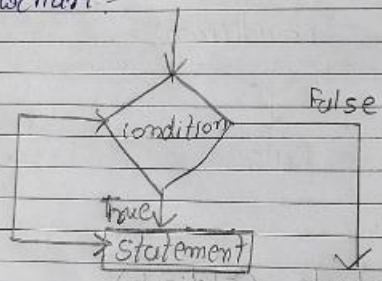
output: ~~4 7 3~~ 1 2 3 4 ... 99 100

⇒ While loop:-

Syntax:

```
while (condition){
        // code
    }
```

flowchart:-



⇒ Example:-

```
#include <stdio.h>
int main () {
 int i=1, n;
 printf ("Enter no: ");
 scanf ("%d", &n);

 while (i<=10) {
  printf ("%d\t", (n*i));
  i++;
 }
 return 0;
}
```

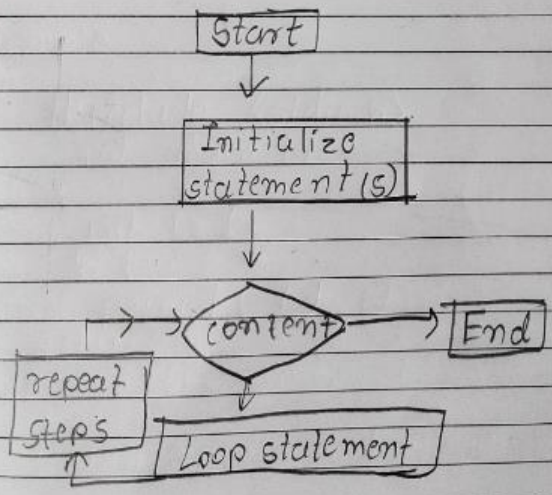
Output : Enter No : 50

50 100 150 200 250 300 350 400

⇒ For Loop:-

Syntax:

```
Initialization; condition;
for (Expression1; Expression2; incre/decre){
        // code to execute.
    }
```



⇒ Example:-

```
#include <stdio.h>
int main () {
   int i, a, b;
   for (i=0, a=12, b=10; i<3; i++);
   {
    printf("%d\t", a+b+i);
   }
   return 0;
}
```

Output : 22 23 24

☆ Find output of the programs :-

1) output - 6

2) i = 0

3) 10

4) 10

5) 11735968 11738112

6) 1

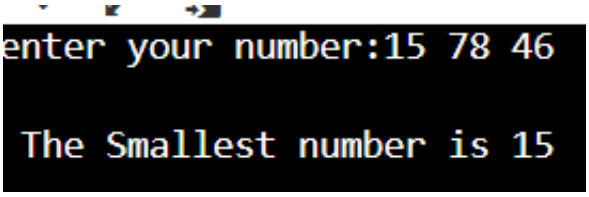
7) 0

1. Make a C program to check least among three integers using Ternary Operator?

Code:

```
#include <stdio.h>
int main()
{
   int num1,num2,num3;
   printf("enter your number:");
   scanf("%d %d %d",&num1,&num2,&num3);
   int temp=((num1<num2)? num1:num2);
   int result=num3<temp?num3:temp;
   printf("\n The Smallest number is %d ",result);
   return 0;
}
```

Output:



2. Make a C Program to find whether a number is: -

a) Perfect Number.

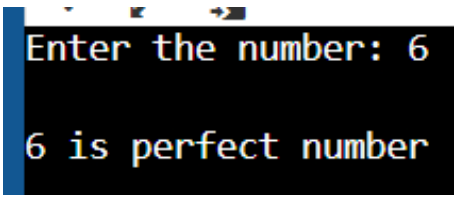
b) Prime Number

c) Even or Odd.

Code: A) perfect number

```
#include <stdio.h>
int main()
{
   int n;
   printf("Enter the number: ");
   scanf("%d",&n);

   int sum = 0;
   for(int i = 1; i < (n-1); i++)
   {
      if((n%i)==0){
         sum=sum+i;
      }
   }
   if(sum==n){
      printf("\n%d is perfect number",n);
   }

   return 0;
}
```

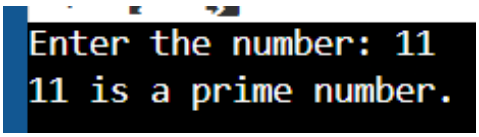
Output:



Code: B) Prime number

```
#include <stdio.h>
int main()
{
   int n;
   printf("Enter the number: ");
   scanf("%d",&n);

   int count = 0;
   for(int i = 2; i < n; i++)
   {
      if(n % i == 0)
         count++;
   }
   if(count == 0)
   {
      printf("%d is a prime number.", n);
   }
   return 0;
}
```

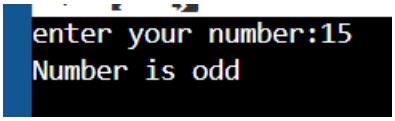
Output:



Code: Even or Odd

```
#include <stdio.h>


int main()
{
   int num,i,flag=1;
   printf("enter your number:");
   scanf("%d",&num);


   if(num%2==0){
      printf("Number is even");
   }
   else{
      printf("Number is odd");
   }
    return 0;
}
```

Output:



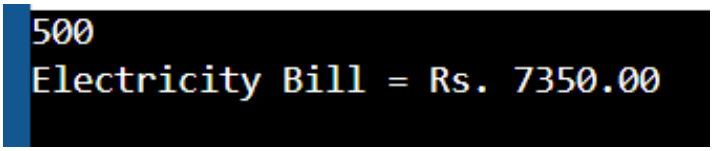
3. Make a program in C to accept monthly electricity consumed unit from user. Find out total monthly electricity charge of a customer as per below given condition.

Criteria of unit charge electricity:
Unit (1 to 100) = Rs. 5 (per unit)
Unit (101 to 200) = Rs. 10 (per unit)
Unit (201 to 300) = Rs. 15(per unit)
Unit (>300) = Rs. 20(per unit)
Tax: 5% of electricity charge.
So, Total electricity charge will be = total unit bill + Tax.

Code:

```
#include <stdio.h>
int main()
{
   int unit;
   float amt,total_amt,tax;
   scanf("%d",&unit);
   if(unit<=100){
      amt=unit*5;
   }
   else if(unit<=200){
      amt=500+((unit-100)*10);
   }
   else if(unit<=300){
      amt=1500+((unit-200)*15);
   }
   else{
      amt=3000+((unit-300)*20);
   }
   tax=amt*0.05;
   total_amt=amt+tax;
   printf("Electricity Bill = Rs. %.2f", total_amt);

   return 0;
}
```

Output:



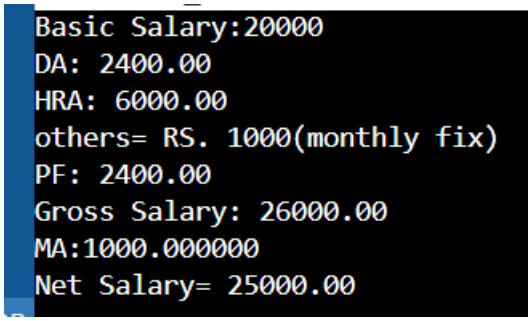
4. Make a program in C to accept basic salary from user. Find out net salary of an employee as per following condition

DA: 12% of Basic salary
HRA: 30% of Basic salary
Others: Rs.1000 (monthly fix)
PF :12% of Basic salary
Gross Salary: (Basic Salary + DA + HRA) - PF
MA: 5% of basic salary
Net Salary = Gross Salary – MA

Code:

```
#include <stdio.h>
int main()
{
   float bs,da,hra,pf,gs,ma,ns;
   printf("Basic Salary:");
   scanf("%f",&bs);
   da=(0.12*bs);
   hra=(0.30*bs);
   pf=0.12*bs;
   gs=(bs+da+hra)-pf;
   ma=(0.05*bs);
   ns=gs-ma;
   printf("DA: %.2f",da);
   printf("\nHRA: %.2f",hra);
   printf("\nothers= RS. 1000(monthly fix)");
   printf("\nPF: %.2f",pf);
   printf("\nGross Salary: %.2f",gs);
   printf("\nMA:%f",ma);
   printf("\nNet Salary= %.2f",ns);

   return 0;
}
```

Output:



5. Make a program in C to accept one five-digit number from user. Find out multiplication and addition of alternate digit.

input: 56789

output:

multiplication=315

addition = 21

Code:

```
#include<stdio.h>

int main(){

   int num,arr[5],i,j,add=0,multi=1,count;
   printf("Enter a Number: ");
   scanf("%d",&num);
   count=num;

   for(i=4;i>=0;i--)
   {
      arr[i]=count%10;
      count=(count-arr[i])/10;

      if(i%2==0)
      {
         add = add + arr[i];
         multi = multi * arr[i];
      }
   }
```

```
    printf("\nAddition of odd digits is: %d",add);
    printf("\nMultiplication of odd digits is: %d",multi);

return 0;

}
```

Output: