Assignment -1

1. WAP for swapping first and last nibbles in given short integert[2 byte] .
Ex . i/p num is 63 .
It's binary : 0000 0000 0011 **1111**
After swap : **1111** 0000 0011 0000

2. WAP to reverse the bits of given character.
Ex. i/p char is 'O' // ascii is 79
It's binary : 0100 1111
After reverse : 1111 0010

3. WAP to find num is divisble by 8 or not .
using bitwise operator + ternary operator
Ex1. i/p 40 o/p : yes
Ex.2 i/p 62 o/p : no

4. WAP to rotate the bits of given short int num.
Ex. i/p num is 31 , rotate bit is 3
It's binary : 0000 0000 0001 1**111**
After rotate : **1110** 0000 0000 0011

5. WAP to delete no of bit from particular position in given number .
Ex. i/p num is 511 , bit is 4 , pos is 2
00000000 00000000 00000001 11**1111**11
after deleting 4 bit fron 2nd pos .
00000000 00000000 00000000 00011111

6. WAP to reverse 1st 6bit to last 6 bit in int.
Ex. i/p number number is 995
00000000 00000000 00000011 11**100011**
after reversing 6 bit only
11000100 00000000 00000011 11000000

7. WAP to set all bit of 1st nibble , clear all bit of 2nd nibble , toggle all bit of 3rd nibble .
Ex. i/p : 0xF5F0 (unsigned short int)
It's binary : 1111 0101 1111 0000
o/p binary : 1111 1010 0000 1111

8. WAP sum of even digit of given number.
[check digit is even or not using bitwise]
Ex. i/p : 7722494 o/p : 12

Assignment -2

1. WAP to print perfect number b/w 1 to 51.

2. WAP to print fibonacii series b/w 0 to 31

3. WAP to print and count palindrome number b/w 51 to 151 .

4. WAP to implement Calculator using switch .

5. WAP print strong number b/w 1 to 251

6. WAP to print and count prime number b/w 51 to 11

7. WAP to print factorial of num b/w 2 to 11.

8. WAP to print armstrong number b/w 1 to 501.

9. WAP to print sum of 1st 4 digit of int num .

10. WAP to print multiplication table from 2 to 9

11. WAP to print 1 st 7 prime number from 21.

12. WAP to print last 5 palindrome num from 99.

13. WAP to count prime digit from given num .
Ex. i/p num is 45678 , o/p is : 2

14. WAP to delete any digit from given num.
Ex. i/p is : 234547 , digit is 4
o/p is : 2357

15. WAP using swich case to check num is prime , perfect , strong , palindrome and armstrong number .

Assignment - 3

---

[1]

```
1
1 3
1 3 5
1 3 5 7
```

[2]

```
2
2 4
2 4 6
2 4 6 8
2 4 6 8 10
```

[3]

```
123456
12345
1234
123
12
1
```

[4]

```
13579
 3579
  579
   79
    9
```

[5]

```
A
A *
A * C
A * C *
A * C * E
```

[6]

```
* * * * * *
1 3 5 7 9
* * * *
1 3 5
* *
1
```

[7]

```
A 1 B 2 C
A 1 B 2
A 1 B
A 1
A
```

[8]

```
5
* *
5 4 3
* * * *
5 4 3 2 1
```

[9]

```
    E
  1 2
C D E
1 2 3 4
A B C D E
```

[10]

```
    #
  A B
  # # #
A B C D
# # # # #
```

[11]

```
1 2 3 4 *
1 2 3 * 5
1 2 * 4 5
1 * 3 4 5
* 2 3 4 5
```

[12]

```
E D C B A
* D C B A
* * C B A
* * * B A
* * * * A
```

[13]

```
1
2   6
3   7   10
4   8   11  13
5   9   12  14  15
```

[14]

```
  1
321
54321
7654321
```

[15]

```
1
1 2
1 2 4
1 2 4 7
1 2 4 7 11
```

[16]

1 2 4 7 11 16
1 2 4 7 11
1 2 4 7
1 2 4
1 2
1

[17]

2
2 3
2 3 5
2 3 5 7
2 3 5 7 11

[18]

1
1 4
1 4 9
1 4 9 16
1 4 9 16 25

[19]

Z
Z Y
Z Y X
Z Y X W
Z Y X W V

[20]

Z
Z Y X
Z Y X W V
Z Y X W V U T

[21]

P Q R S T U V
P Q R S T
P Q R
P Q
P

[22]

* * * * *
4 3 2 1
* * *
2 1
*

[23]

4 3 2 1
A B C
2 1
A

[24]

A
B B
C C C
D D D D
E E E E E

[25]

F E D C B A
E D C B A
D C B A
C B A
B A
A

[26]

1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4

[27]

A
C B A
E D C B A
G F E D C B A
I H G F E D C B A

[28]

*****
* *
* *
* *
*****

[29]

*
***
*****
****

[30]

A
A B
A B C
A B C D
A B C D E
A B C D
A B C
A B
A

ARRAY

NOTE: take array i/p at runtime

1.
WAP in C to reverse the element of given array

i/p: int a[6]={2,3,4,5,6,7};
o/p: 7 6 5 4 3 2

2.
WAP in C to delete a element at desired position from an array .

i/p: int a[6]={2,3,4,5,6,7}; , pos= 2
o/p: 2 3 5 6 7

3.
WAP in C to insert an element at desired postion in an array .

i/p : int a[6]={2,3,4,5,6}; ,pos=1,n=9
o/p : 2 9 3 4 5 6

4.
WAP in C delete duplicate element of array .

i/p: int a[7]={2,2,2,3,3,3,4};
o/p : 2 3 4

5.
WAP in C to count duplicate elements .

i/p: int a[8]={1,1,2,3,2,2,1,7};
o/p: 1 -->3 times , 2-->3 times

6.
WAP in C to print non-repeted elemnts.

i/p: int a[8]={1,1,2,3,5,2,1,7};
o/p: 3 5 7

7.
WAP in C to print second largest element.

i/p: int a[5]={1,21,51,21,11};
o/p: 21

8.
WAP in C to print perfect num in array.

i/p: int a[5]={2,6,6,28,11};
o/p: 6 6 28 , count= 3

9.
WAP in C to print strong num in array.

i/p :int a[6]={2,2,3,4,145,6}
o/p : 2 2 145 , count = 3

10.
WAP in C to write bubble sort ,selection sort.

STRING

NOTE: take string i/p at runtime

1.
WAP in C to count no of words in given string.

i/p: s[20]="abc pqr xyz 12"
o/p: word count= 4

2.
WAP in C to count vowels in string.

i/p: s[20]="abc pqr aeio"
o/p: vowels count= 5

3.
WAP in C to check string is Palindrome or not.

i/p: s[20]= "radar"
o/p: yes

4.
WAP in C to delete desired char from string.

i/p: s[20]="abcccab" , char= c
o/p: abab

5.
WAP in C to remove conjucutive spaces in sring

i/p: s[20]="abc coding sirji"
o/p: abc coding sirji

6.
WAP in C to delete duplicate char from string.

i/p: s[20]="abc abc ppp 122"
o/p : abc p12

7.
WAP in C to count dublicate char from string.

i/p: s[20]="aaababcdeb"
o/p: a-->4 times b-->3 times

8.
WAP in C to sort string in any order .

i/p : "615DSppzaA"
o/p: 156ADSappz

9.
WAP in C to reverse all words in string.

i/p: "coding sirji vector"
o/p: gnidoc ijris rotcev

10.
WAP in C to merge 2 string to another string.
i/p: s1[10]="1234" s2[10]="ABCD"
o/p : s3[20]="1A2B3C4D"

FUNCTION

1.

WAP in C using function to sum of digit of all elements in array , store results in another array.

i/p: int a[6]={1,22,121,34,78,444};
o/p: int b[6]= {1, 4, 4, 7, 15, 12 };
void sum_fun(int *a,int *b , int ele);

2.

WAP in C using function to reverse all elements of array ,store results in another array.

i/p: int a[6]={12,42,123,34,78,414};
o/p: int b[6]={21,24,321,43,87,414};
void rev_fun(int *a,int *b , int ele);

3.

WAP in C using function to delete 1st digits of all elements in array .

i/p: int a[6]={12,142,1234,314,78,414};
o/p: int a[6]={2,42,234,34,8,14};
void del_fun(int *a, int ele);

4.

WAP in C using function to count strong and armstrong number elements in array .

i/p: int a[6]={2,153,145,2,3,153};
o/p: strong number count = 3
armstrong number count = 5
int strong_fun(int *a, int ele);
int armstrong_fun(int *a, int ele);

5.

WAP in C using function to count -ve element (bitwise op) and delete -ve elements in array .

i/p: int a[6]={-2, 2,-5,-12,5,-7};
o/p: -ve number count = 4
int a[6]= {2,5};
int count_del_fun(int *a, int ele);

6.

WAP in C using function to right rotate array 2 times .

i/p: int a[6]={-2, 2,-5,-12,5,-7};
o/p: int a[6]={5,-7,-2, 2,-5,-12};
void rotate_fun(int *a, int ele ,int n);

7.

WAP in C using function to insert num in array at given particular location (index).

i/p: int a[6]={-1,2,3,-5,-7}; , n= 99 , p=2
o/p: int a[6]= {-1, 2 ,**99** ,3,-5, -7};
void in_fun(int *a, int ele,int n, int p);

8.

WAP in C using function to merge 2 array data in 3rd array .

i/p: int a[6]={2,3,4}, b[3]={11,22,33};
o/p: int c[6]= {2,11,3,22,4,33};

void in_fun(int *a, int *b, int *c,
int ele1,int ele2, int ele3);

9.

WAP in C using function to **delete the duplicate** char from given string .

i/p: char s[20]="abcaaabbccaa";
o/p: abc
void del_fun(char * s);

10.

WAP in C using function to **delete all** digits in strings and count deleted digits .

i/p: char s[20]="a1b2c3d4123";
o/p: abcd , digit count = 7
int del_count_fun(char * s);

11.

WAP in C using function to revesre all word in string ,count no of word having digits .
i/p: char s[20]="coding sirji vec123 A123 ";
o/p: gnidoc ijris **321cev 321A**
word count = 2

void rev_word_fun(char * s);
int count_word_fun(char * s);

Note: Take all input at runtime (use scanf).
Use same function prototype only .

Recursive Function

1.
WAP in C using Recursive function to sum of even digits of given any int number .
i/p: n= 2345 o/p: sum= 6
int rec_fun_sum(int num);

2.
WAP in C using Recursive function to count digit less than 6 of given any int number .
i/p: n= 2658942 o/p: count= 4
int rec_fun_count(int num);

3.
WAP in C using Recursive function to product of digit factor of 3 given any int number .
i/p: n= 345638 o/p: product= 54
int rec_fun_product(int num);

4.
WAP in C using Recursive function to sum of last 3 digits of given any int number .
i/p: n= 23456 o/p: sum= 15
int rec_fun_sum(int num , int c);

5.
WAP in C using Recursive function to reverse the number of given any int number .
i/p: n= 23456 o/p: rev = 65432
int rec_fun_rev(int);

6.
WAP in C using Recursive function to check given num is perfect or not .
i/p: n= 6 o/p: yes perfect
int rec_fun_perfect(int);

8.
WAP in C using Recursive function to count set bit in given num. // pass address of variable
i/p: n= 63 o/p: count: 6
int rec_fun_count(int *);

7.
WAP in C using Recursive function to check given num is prime or not .
i/p: n= 17 o/p: yes prime
int rec_fun_prime (int ,int);

8.

WAP in C using Recursive function to count array element less than 99 more than 39.
i/p: int a[6]={71,53,145,21,49,153};
o/p: count = 3
int rec_fun_count_arr(int *p, int ele);

9.
WAP in C using Recursive function to sum of half of array element.
i/p: int a[6]={10,20,30,44,55,66};
o/p: sum = 60
int rec_fun_sum_arr(int *p, int ele);

10.
WAP in C using Recursive function to reverse array elements and print array in main .
i/p: int a[6]={11,22,33,44,55,66};
o/p: a[6]={66,55,44,33,22,11};
void rec_fun_rev_arr(int *p, int *q);

10.
WAP in C using Recursive fun to reverse string .
i/p: char s[20]="123 abc 789";
o/p: 987 cba 321
void rec_fun_rev_string(char *p, char *q);

11.
WAP in C using Recursive function to count char in given any string .
i/p: char s[20]="123 aacc tata "; , ch= 'a'
o/p: count = 4
int rec_fun_count_string(char *p, char ch);

12.
WAP in C using Recursive fun to reverse bits .
i/p : n= 31
00000000 00000000 00000000 00011111
o/p: // print binary in main function
11111000 00000000 00000000 00000000
int rec_fun_rev_bit (int);
void rec_fun_binary (int);

WAP in C using Recursive function given fun.

13. copy a string
void my_strncpy(char *p, char *q, int n);

14. compare two strings
int my_strncmp(char *p, char *q, int n);

15. locate character in string
char* my_strchr (char *p, char ch);

Command Line Arguments [CLA]

int main(int argc , char **argv)

1.
WAP in C using CLA to write given function.

int my_atoi(const char *nptr);

2.
WAP in C using CLA to print 1st digit of num.

i/p: ./a.out 1234
o/p : 1

3.
WAP in C using CLA to prime number b/w 11 to 66 .

i/p: ./a.out 11 66

4.
WAP in C using CLA to print strong number b/w 1 to 199 .

i/p: ./a.out 1 199

5.
WAP in C using CLA to take array input and print in reverse order .

i/p: ./a.out 11 22 33 44 55
o/p : 55 44 33 22 11

6.
WAP in C using CLA to write given function.

double my_atof(const char *nptr);

7.
WAP in C using CLA to print average and sum of 3 float number .
i/p: ./a.out 12.56 45.7 345.23
o/p : sum= 403.49 avg=134.496

8.
WAP in C using CLA to implement Calculator
ex1 ./a.out 44 + 200
o/p : 244
ex2 ./a.out 44 / 0
o/p : FPE

9.
WAP in C using CLA to reverse string .

i/p : ./a.out "123 789 CBA"
o/p: ABC 987 321

10.
WAP in C using CLA to insert one char in string at given postion .
i/p : ./a.out ABCDEF 2 P
o/p : ABPCDEF

m

11.
WAP in C using CLA to print given Pattern .

i/p: ./a.out 5

```
9 7 5 3 1
  7 5 3 1
    5 3 1
      3 1
        1
```

./a.out 3

```
5 3 1
  3 1
    1
```

12.
WAP in C using CLA to count digit in string.

i/p: ./a.out gd53gd82js
o/p: count= 4

13.
WAP in C using CLA to write given functio.

i/p: ./a.out coding c_ds
o/p : len1= 6 , len2= 4
both string not equal

int my_strlen(char *);
int my_strcmp(char *, char*);
int main(int argc , char **argv) ;

FILE

1.
WAP in C to count line , word and digit in file.
i/p: 123 abc coding
file c ds 789 sirji

o/p: no of line = 2 , word= 8 , digit= 6
-------$ ./a.out data

2.
WAP in C to convet small to capital later .
Using fseek function .
--------$ ./a.out data

3.
WAP in C to print all word length .
i/p: 123 abc coding
file sirji FILE
o/p : 3 3 6 4 5 4
-------$ ./a.out data

4.
WAP in C merge char by char in 3rd file .
---------$ ./a.out data1 data2 data3

5.
WAP in C to merge word by word in 3rd file .
---------$ ./a.out data1 data2 data3

6.
WAP in C to merge line by line in 3rd file.
---------$ ./a.out data1 data2 data3

7.
WAP in C to delete any line of given file .
--------$ ./a.out data line_no

8.
WAP in C to delete line 1st and last line only in given any file using CLA.
--------$ ./a.out data

9.
WAP in C to replace one word with another word.
-------$ ./a.out data hello coding

10,
WAP in C to convert all word 1st and last char as a capital in file using CLA.
i/p: 123 abc coding
file ds 789 sirji

o/p: 123 AbC CodinG
FilE DS 789 SirjI
-------$ ./a.out data

PROJECT

Title : Preprocessor

-------$ vi project.c

```
int main(){

// write logic hare
// Task1 + Task2 + Task3
}
```

-------$ cc project.c -o my_Preprocessor
-------$ ./my_Preprocessor abc.c

------$ vi abc.c // input file

```
#include<stdio.h>
#include<string.h>
#define abc 3456
#define pf printf
#define coding 65
int main(){
// delete single line comment
int k=abc;
/* remove multi line comment
*/
pf("%d %d\n",k,coding );
}
```

-----$ vi abc.i // output file

```
header file content

int main(){

int k= 3456;

printf("%d %d \n",k, 65 );
}
```

Task1: **Remove All the Comments**

Task2: **Header File Inclusion**

Task3: **Macro Substitution**

NOTE: use Modular coding with makefile

ID:

Name:

1. WAP in C to print all factor of 512.

2. WAP in C to print all odd factor of 900

3. WAP in C to print even number (using bitwise operator) b/w 299 to 275.

4. WAP in C to print alternate odd number between 21 to 51.

5. WAP in C to print given series .

1 2 4 7 11 16 22

6. WAP in C to print given series .

-13 -8 -3 2 7 12 17

7. WAP in C to print given series .

2 9 28 65 126 217

NOTE: Use single for loop

Name: **ID:**

7 1.WAP in C to count digits in b/w 4 to 7

- i/p: n= 3424697 o/p: c= 4

2.WAP in C : product of last 2 digit.

i/p: n= 3457

o/p: p= 35 (5*7)

3.WAP in C to print sum of last 3 digit.

i/p: 76543

o/p: 12 (3+4+7)

h 4.WAP in C to print sum of digits which are divisible by 3 .

i/p: 345673

o/p: 12 (3+6+3)

5. .WAP in C : sum of alternate digits.

Ex1 i/p: 3456 o/p: 8 (3+5)

Ex2 i/p: 12345 o/p: 9 (1+3+5)

ID:V23BE2___ **Name :**

WAP in C to print sum of first 4 odd digits .
Note : print result in decimal and binary .

Ex1 n= 123789456 o/p= 20 (1+3+7+9)
00000000 00000000 00000000 00010100

ID:V23BE___ Name :

ID:V23BE2___ Name :

WAP in **ARRAY** to keep -ve element left side and +ve element on right side in same array .

i/p: int a[6]={11,-22,-33,44,55,-66};

o/p: -22 , -33 , -66 , 11, 44, 55

Done

ID:V23BE___ Name :

WAP in C to reverse word having digit in **string**

i/p: ab2 @#$ v23be1 ABC cs 123
o/p: 2ba @#$ 1eb32v ABC cs 321
void word_rev_fun(char *);

8+4-(2+1)

* 3%.5+2019

8+4-(2+1)*3%.5+2019

~~20-19~~

8+4-3*3%.5+2019

8+4-9%.5+2019

8+4-4+2

8+0+2

12

17%6-5/2+

(6*3-5)/5

17%.6-5/2+(6*3-5)/5

17%.6-5/2+13/5

5-4+2

1+2 (3)

2 5/2
2|5
4
1 2
6|17
12
0 5
2
5|13
10

[OPERATOR]

1.
```
int main(){
int a=11,b=7,c;
c= b%a;
printf("%d \n",c); }
```
2.
```
int main(){
int a=11,b=7,c;
c= b%a/3;
printf("%d \n",c); }
```
3.
```
int main(){
int a=11,b=7,c;
c= -a+b;
printf("%d \n",c); }
```
4.
```
int main(){
int a=17,b=7,c;
c= a%10*10;
printf("%d \n",c); }
```
5.
```
int main(){
int a=17,b=7,c;
c= a>-b;
printf("%d \n",c);}
```
6.
```
int main(){
int a=17,b=7,c;
c= a>-b;
printf("%d \n",c);}
```
7.
```
int main(){
int a=17,b=7,c;
c= -a>=b;
printf("%d \n",c);
}
```
8.
```
int main(){
int a=17,b=7,c;
c= 7%(-a>=b);
printf("%d \n",c);}
```
9.
```
int main(){
int a=17,b=7,c;
c= a%(b-7);
printf("%d \n",c);
}
```
10.
```
int main(){
int a=17,b=7,c;
c= a%(b-7);
printf("%d \n",c);}
```

erka G'/.0 lena

11.
```
int main(){
int a=17,b=7,c;
c= (a-b)*10;
printf("%c \n",c);}
```
12.
```
int main(){
int a=17,b=7,c;
c= b<a/a+b;
printf("%d \n",c);}
```
13.
```
int main(){
float a=17,b=7,c;
c= a%b+10;
printf("%f \n",c);}
```
14.
```
int main(){
float a=17,b=7,c;
c= 35/10;
printf("%f \n",c);}
```
15
```
int main(){
float a=17,b=7,c;
c= 'A'/10;
printf("%.3f \n",c);}
```
16.
```
int main(){
float a=17,b=10,c;
c= 'a'/b;
printf("%.2f \n",c);}
```
17.
```
int main(){
double a=17,b=10,c;
c= 'b'/b;
printf("%lf \n",c);}
```
18.
```
int main(){
double a=17,b=10,c;
c= 'b'% b;
printf("%lf \n",c);
}
```
19.
```
int main(){
int a=0,b=10,c;
c= a||(b=0);
printf("%d %d\n",c,b);}
```
20.
```
int main(){
int a=0,b=10,c;
c= (a==0)||(b=0);
printf("%d %d\n",c,b);
}
```

21.
```
int main(){
int a=0,b=10,c;
c= (a==0)&&(b=0);
printf("%d %d\n",c,b);}
```
22.
```
int main(){
int a=0,b=10,c=4,d;
d= (a==0)&&(b=0)||(c=5);
printf("%d %d %d\n",b,c,d);}
```
23.
```
int main(){
int a=0,b=10,c=4,d;
d= (a==0)&&(b=2)||(c=5);
printf("%d %d %d\n",b,c,d);}
```
24.
```
int main(){
int a='A';
!a;
printf("%d \n",a);}
```
25.
```
int main(){
int a=0,b=10,c=4,d;
d= (a==0)||(b=2)||(c=5);
printf("%d %d %d\n",b,c,d);}
```
26.
```
int main(){
int a=0,b=300,c=4,d;
d= (a==0)||(b=2)&&(c=5);
printf("%d %d %d\n",b,c,d);}
```
27.
```
int main(){
int a=1,b=10,c=400,d;
d= (a==0)&&(b=2)&&(c=5);
printf("%d %d %d\n",b,c,d);}
```
28.
```
int main(){
int  vector=1=2;
printf("%d \n",vector);}
```
29.
```
int main(){
int  coding='1'+3;
printf("%x\n",coding);}
```
30.
```
int main(){
int  sirji=0x55;
sirji= 0xAA+sirji;
printf("%d\n",sirji);
}
```

INT + SHORT INT

1.
```
int main(){
short int z=99;
printf("%c\n",z);
}
```
2.
```
int main(){
short int z=919;
printf("%o\n",z);
}
```
3.
```
int main(){
short int z=0777;
printf("%d\n",z);
}
```
4.
```
int main(){
short int z=511;
printf("%x\n",z);
}
```
5.
```
int main(){
short int z=0511;
printf("%x\n",z);
}
```
6.
```
int main(){
short int i='0';
printf("%x\n",i*i);
}
```
7.
```
int main(){
short int i=0x12345;
printf("%x\n",i);
}
```
8.
```
int main(){
short int i=0x12345;
printf("%d\n",i);
}
```
9.
```
int main(){
short int i=0x12345;
printf("%o\n",i);
}
```
10.
```
int main(){
short int i=204;
char c= i;
printf("%d\n",c);
}
```
11.
```
int main(){
short int i=01234;
char c= i;
printf("%d\n",c);
}
```
12.
```
int main(){
short int i=1000;
char c= i;
printf("%d\n",c);
}
```
13.
```
int main(){
short int i=1010;
unsigned char c= i;
printf("%d\n",c);
}
```
14.
```
int main(){
short int i=0xFACE;
unsigned char c= i;
printf("%d\n",c);
}
```
15.
```
int main(){
short int i=0x98765;
unsigned char c= i;
printf("%o\n",c);
}
```
16.
```
int main(){
int  k=0x1234567;
short int i=k;
unsigned char c=i;
printf("%x\n",c);
}
```
17.
```
int main(){
int  k=0x1234567;
short int i=k;
unsigned char c=i;
printf("%o\n",i);
}
```
18.
```
int main(){
int  k=0x6789;
short int i=k;
char c=i;
printf("%d\n",c);
}
```
19.
```
int main(){
int  k=012345;
short int i=k;
char c=i;
printf("%d\n",c);
}
```
20.
```
int main(){
int  k=-500;
short int i=k;
char c=i;
printf("%d\n",c);
}
```
21.
```
int main(){
int  k=-'d';
short int i=k;
char c=i+10;
printf("%d\n",c);
}
```
22.
```
int main(){
int  k=0x111;
short int i=0111;
char c=111;
   k= k+c+i;
printf("%d\n",k);
}
```
23.
```
int main(){
int  k=0x111;
short int i=0111;
char c=111;
   c= k+c+i;
printf("%d\n",c);
}
```
24.
```
int main(){
unsigned int k=400;
short int i=(char)k;
printf("%d\n",i);
}
```
25.
```
int main(){
unsigned int k=304;
short int i=(char)k;
printf("%c\n",i);
}
```

ID :
NAME:

Assignment -1

1. WAP for swapping first and given short integer[2 byte] .
Ex . i/p num is 63 .
It's binary : 0000 0000 0011
After swap : 1111 0000 0011

2. WAP to reverse the bits of
Ex. i/p char is 'O' // ascii is
It's binary : 0100 1111
After reverse : 1111 0010

3. WAP to find num is divisi
using bitwise operator +
Ex1. i/p 40 o/p : y
Ex.2 i/p 62 o/p : r

4. WAP to rotate the bits
Ex. i/p num is 31 , rota
It's binary : 0000 00
After rotate : 1110 00

5. WAP to delete no of
position in given numb
Ex. i/p num is 511 ,
00000000 00000000
after deleting 4 bit
00000000 00000000

6. WAP to reverse
Ex. i/p number nu
00000000 000000
after revers
11000100 00000

7. WAP to set al
of 2nd nibble ,
Ex. i/p : 0xF
It's binary : 1
o/p binary : 1

8. WAP sum
[check digit
Ex. i/p : 7

Number System .

1.
```
void main(){
char ch = 'A';
printf("%d\n",ch);
}
```
65

2.
```
void main(){
char ch = 'd';
printf("%d\n",ch);
}
```
100

3.
```
void main(){
char ch = '3';
printf("%d\n",ch);
}
```
51

4.
```
void main(){
char ch = 50;
printf("%c\n",ch);
}
```
2

5.
```
void main(){
char ch = 99;
printf("%c\n",ch);
}
```
c

6.
```
void main(){
char ch = 85;
printf("%d\n",ch);
}
```
U

7.
```
void main(){
char ch = 85;
printf("%d\n",ch);
}
```
U

8.
```
void main(){
char ch = 0x12;
printf("%d\n",ch);
}
```
18

9.
```
void main(){
char ch = 0x48;
printf("%d\n",ch);
}
```
72

10.
```
void main(){
char ch = 0x2F;
printf("%d\n",ch);
}
```
47

11.
```
void main(){
char ch = 11;
printf("%d\n",ch);
}
```
11

12.
```
void main(){
char ch = 0x11;
printf("%d\n",ch);
}
```
17

13.
```
void main(){
char ch = 011;
printf("%d\n",ch);
}
```
9

14.
```
void main(){
char ch = 0111;
printf("%d\n",ch);
}
```
73

15.
```
void main(){
char ch = 055;
printf("%d\n",ch);
}
```
45

16.
```
void main(){
char ch = 029;
printf("%d\n",ch);
}
```
Error

17.
```
void main(){
char ch = 0x29;
printf("%d\n",ch);
}
```
41

18.
```
void main(){
char ch = 200;
printf("%d\n",ch);
}
```
55

19.
```
void main(){
char ch = 300;
printf("%d\n",ch);
}
```
45

20.
```
void main(){
char ch = 0x123;
printf("%d\n",ch);
}
```
39

21.
```
void main(){
char ch = 0x101;
printf("%d\n",ch);
}
```
1

22.
```
void main(){
char ch = 0101;
printf("%c\n",ch);
}
```
A

23.
```
void main(){
char ch = 'a';
printf("%o\n",ch);
}
```
141

24.
```
void main(){
char ch = 'e';
printf("%x\n",ch);
}
```
145

25.
```
void main(){
char ch = '0';
printf("%x\n",ch);
}
```
30

26.
```
void main(){
char ch = A;
printf("%c\n",ch);
}
```
Error

27.
```
void main(){
char ch = 0xA;
printf("%d\n",ch);
}
```
10

28.
```
void main(){
char ch = 0x1070;
printf("%d\n",ch);
}
```
120

29.
```
void main(){
char ch = 141;
printf("%d\n",ch);
}
```
-115

30.
```
void main(){
char ch = 241;
printf("%d\n",ch);
}
```
-15

31.
```
void main(){
char ch = 291;
printf("%d\n",ch);
}
```
35

Assignm

1. WAP for swappin
given short integer[2
Ex . i/p num is 63
It's binary : 0000 0(
After swap : 1111 0(

2. WAP to reverse th
Ex. i/p char is 'O' /
It's binary : 010(
After reverse : 1111

3. WAP to find num i
using bitwise oper
Ex1. i/p 40 o/p
Ex.2 i/p 62 o/p

4. WAP to rotate the
Ex. i/p num is 31
It's binary : 00(
After rotate : 111

5. WAP to delete n
osition in given n
x. i/p num is 51
0000000 000000
after deleting 4 l
000000 00000(

WAP to revers
i/p number
00000 00000
after rever
00100 0000(

AP to set al
d nibble ,
i/p : 0xF
nary : 1
nary : 1

sum (
digit i
: 77

Char data type

Name :
ID :

1
```
void main(){
char CSK=0xCC;
printf("%d \n",CSK);
}
```
-52

2
```
void main(){
char CSK=0xFB;
printf("%d \n",CSK);
}
```
-5

3
```
void main(){
char CSK=0xDCBA;
printf("%d \n",CSK);
}
```
-70

4
```
void main(){
char CSK=0xcab;
printf("%d \n",CSK);
}
```
-85

5
```
void main(){
char CSK=0x0F;
printf("%d \n",CSK);
}
```
15

6
```
void main(){
char RCB=0770;
printf("%d \n",RCB);
}
```
-8

7
```
void main(){
char RCB=0321;
printf("%d \n",RCB);
}
```
-47

8
```
void main(){
char RCB=0333;
printf("%d \n",RCB);
}
```
-37

9
```
void main(){
char RCB=0300;
printf("%d \n",RCB);
}
```
-64

10
```
void main(){
char RCB=03232;
printf("%d \n",RCB);
}
```
-1026

11
```
void main(){
char SRH=145;
printf("%d \n",SRH);
}
```

12
```
void main(){
char SRH=185;
printf("%d \n",SRH);
}
```

13
```
void main(){
char SRH=225;
printf("%d \n",SRH);
}
```
-31

14
```
void main(){
char SRH=355;
printf("%c \n",SRH);
}
```

15
```
void main(){
char SRH=455;
printf("%d \n",SRH);
}
```

16
```
void main(){
unsigned char coding;
coding = 0x12345;
printf("%d \n",coding);
}
```

17
```
void main(){
unsigned char coding;
coding = 0xAC;
printf("%d \n",coding);
}
```

18
```
void main(){
unsigned char coding;
coding = 0xCC;
printf("%d \n",coding);
}
```

19
```
void main(){
unsigned char coding;
coding = 0x786;
printf("%d \n",coding);
}
```

20
```
void main(){
unsigned char coding;
coding = 0xBBC;
printf("%d \n",coding);
}
```

21
```
void main(){
unsigned char sirji;
sirji = 500;
printf("%d \n",sirji);
}
```

22
```
void main(){
unsigned char sirji;
sirji = 1010;
printf("%d \n",sirji);
}
```

23
```
void main(){
unsigned char sirji;
sirji = 1023;
printf("%d \n",sirji);
}
```

24
```
void main(){
unsigned char sirji;
sirji = -100;
printf("%d \n",sirji);
}
```

25
```
void main(){
unsigned char sirji;
sirji = -'A';
printf("%d \n",sirji);
}
```

Assignmen

1. WAP for swapping f
given short integer[2 by
Ex . i/p num is 63 .
It's binary : 0000 000(
After swap : 1111 000(

2. WAP to reverse the
Ex. i/p char is 'O' //
It's binary : 0100
After reverse : 1111

3. WAP to find num i
using bitwise oper
Ex1. i/p 40 o/p
Ex.2 i/p 62 o/p

4. WAP to rotate th
Ex. i/p num is 31
It's binary : 00
After rotate : 11

5. WAP to delete
position in given
Ex. i/p num is
00000000 0000
after deleting
00000000 000(

6. WAP to rev
Ex. i/p numb
00000000 00
after re
11000100 0(

7. WAP to s
f 2nd nibb
x. i/p :
's binary
p binary

WAP s
eck di
i/p

ID: V23bE2L3
Name: Harshil

1
```
int main(){
char k=0xf1,a;
a=k<<2;
printf("%d \n",a);
}
```

2.
```
int main(){
char k=0xf7,a;
a=k<<04;
printf("%d \n",a);
}
```

3.
```
int main(){
short int f=0310;
char k=0xf8,a;
a=k&f;
printf("%d \n",a);
}
```

4.
```
int main(){
short int f=0xF;
char k=63,a;
a=k^f;
printf("%c \n",a);
}
```

5.
```
int main(){
unsigned short int f;
 f= 0xffff;
char k=f;
printf("%d \n",k);
}
```

6.
```
int main(){
printf("%d \n",(char)149);
}
```

7.
```
int main(){
int k=-4?-4:1;
printf("%x\n",k);
}
```

8.
```
int main(){
int k=-1023;
printf("%x\n",k);
}
```

9.
```
int main(){
int k=sizeof(0xC2);
int c=k|0x2EC;
printf("%d\n",c);
}
```

10.
```
int main(){
int k=-71;
printf("%d\n",k>>2);
}
```

ID:
Name:

[POINTER]

1.
```
 int main(){
int a=10;
int *p=&a;
printf("%d \n",*p);
}
```
2.
```
int main(){
int a=100;
int *p= &a;
printf("%x \n",*p);
}
```
3.
```
int main(){
int a='1';
int *p= &a;
printf("%o \n",*p);
}
```
4.
```
int main(){
int a=0x10;
int *p= &a;
printf("%d \n",*p);
}
```
5.
```
int main(){
int a=1000;
int *p=&a;
printf("%x \n",*p);
}
```
6.
```
int main(){
int a=2000;
int *p= &a;
printf("%d \n",*p+10);
}
```
7.
```
int main(){
int a=100;
int *p= &a;
printf("%c \n",*p-3);
}
```
8.
```
int main(){
int a=100;
int *p= &a;
printf("%d\n",*p+*p);
}
```
9.
```
int main(){
int a=0101;
int *p= &a;
printf("%d\n",*p);
}
```
10.
```
int main(){
int a=100;
int *p= &a;
printf("%d\n",'a'-*p);
}
```
11.
```
int main(){
int a=10;
int *p= &a;
*p=200;
printf("%d\n",a+2);
}
```
12.
```
int main(){
int a=10;
int *p= &a;
*p=20+a;
printf("%d\n",a);
}
```
13.
```
int main(){
int a=10;
int *p= &a;
*p=20+'0';
printf("%d\n",*p%10);
}
```
14.
```
int main(){
int a=10;
int *p= &a;
*p=20+'0';
printf("%d\n",a+*p*10);
}
```
15.
```
int main(){
int a=10;
char *p=(char*)&a;
printf("%d\n",*p);}
```
16.
```
int main(){
int a=100;
char *p=(char*)&a;
printf("%c \n",*p);
}
```
17.
```
int main(){
int a=110;
char *p=(char*)&a;
printf("%x \n",*p);
}
```
18.
```
int main(){
int a=0123;
char *p=(char*)&a;
printf("%d \n",*p);
}
```
19.
```
int main(){
int a=130;
char *p=(char*)&a;
printf("%d\n",*p);
}
```
20.
```
int main(){
int a=148;
char *p=(char*)&a;
printf("%d\n",*p);
}
```
21.
```
int main(){
int a=198;
char *p=(char*)&a;
printf("%d\n",*p);
}
```
22.
```
int main(){
int a=220;
char *p=(char*)&a;
printf("%d\n",*p);
}
```
23.
```
int main(){
int a=290;
char *p=(char*)&a;
printf("%d\n",*++p);}
```
24.
```
int main(){
int a=479;
char *p=(char*)&a;
printf("%d\n",*p);
}
```
25.
```
int main(){
int a=581;
char *p=(char*)&a;
printf("%c\n",*++p);}
```

Name:

ID---> V23BE2 ___

1.
WAP in C to print last digit of given any int num.
Ex1. i/p : 23456 o/p : 6

2.
WAP in C to sum of digit of 2 digit number only.
Ex1. i/p : 56 o/p : 11

3.
WAP in C to reverse num of 2 digit number only.
Ex1. i/p : 12 o/p : 21

4.
WAP in C , take 2 number , sum of 1st num 1st digit and 2nd num last digit.
Ex1. i/p n1 = 12 , n2 = 46
o/p: 1+6 // 7

5.
WAP in C to sum of digit of 3 digit number only.
Ex1. i/p : 123 o/p : 6

6.
WAP in C for given basic **Math Formula .**

$(a + b)^2 = a^2 + 2ab + b^2$

Ex1. i/p : a = 2 , b= 3 o/p : 25

7.
WAP in C for given basic **Math Formula .**

$(a + b)3 = a3 + 3a2b + 3ab2 + b3$

Ex1. i/p : a = 2 , b= 4 o/p : 216

NOTE: Don't use any loop in above all 7 program . Use only operator.