Twisters 1

What will be output if you will compile and execute the following c code?

```c
void main(){
 int a=-12;
 a=a>>3;
 printf("%d",a);
}
```
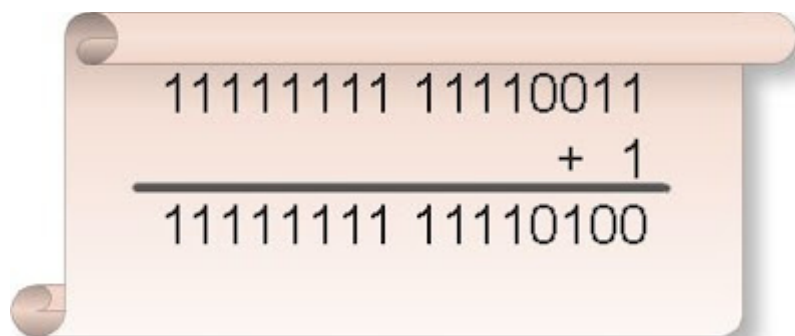
(a) -4
(b) -3
(c) -2
(d) -96
(e) Compiler error

Answer :(c)
Explanation:
Binary value of 12 is: 00000000 00001100
Binary value of -12 wills 2's complement of 12 i.e.

So binary value of -12 is: 11111111 11110100

| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

Since it is negative number so output will
also a negative number but its 2's
complement.

Hence
final

```
00000000 00000001
         +   1
00000000 00000010
```

out put will be:

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

And its decimal value is: 2
Hence output will be:-2

What will be output if you will compile and
execute the following c code?
```c
void main(){
 int a,b;
 a=1,3,15;
 b=(2,4,6);
 clrscr();
 printf("%d ",a+b);
 getch();
}
```
(a) 3
(b) 21
(c) 17
(d) 7
(e) Compiler error

Answer: (d)

Explanation:

In c comma behaves as separator as well as operator.

a=1, 3, 15;

b= (2, 4, 6);

In the above two statements comma is working as operator. Comma enjoys least precedence and associative is left to right.

Assigning the priority of each operator in the first statement:

What will be output if you will compile and execute the following c code?

```c
void main(){
 int i=0;
 if(i==0){
  i=((5,(i=3)),i=1);
  printf("%d",i);
 }
 else
  printf("equal");
}
```

(a) 5
(b) 3
(c) 1
(d) equal
(e) None of above

Answer: (c)

What will be output when you will execute following c code?

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

```c
int main(){
  signed x;
  unsigned y;
  x = 10 +- 10u + 10u +- 10;
  y = x;
  if(x==y)
    printf("%d %d",x,y);
  else if(x!=y)
    printf("%u  %u",x,y);
  return 0;
}
```

Choose all that apply:

(A) 0 0

(B) 65536 -10

(C) 0 65536

(D) 65536 0

(E) Compilation error

Turbo C++ 3.0: 0 0
Turbo C ++4.5: 0 0
Linux GCC: 0 0
Visual C++: 0 0

Consider on the expression:
x = 10 +- 10u + 10u +- 10;
10: It is signed integer constant.
10u: It is unsigned integer constant.
X: It is signed integer variable.
In any binary operation of dissimilar data type for example: a + b
Lower data type operand always automatically type casted into the operand of higher data type before performing the operation and result will be higher data type.
As we know operators enjoy higher precedence than binary operators. So our expression is:
x = 10 + (-10u) + 10u + (-10);
 = 10 + -10 + 10 + (-10);
 = 0
Note: Signed is higher data type than unsigned int.
So, Corresponding signed value of unsigned 10u is +10

What will be output when you will execute following c code?

```c
#include<stdio.h>
int main(){
  int a= sizeof(signed) +sizeof(unsigned);
  int b=sizeof(const)+sizeof(volatile);
  printf("%d",a+++b);
  return 0;
}
```

Choose all that apply:
(A) 10
(B) 9
(C) 8
(D) Error: Cannot find size of modifiers
(E) Error: Undefined operator +++

Turbo C++ 3.0: 8
Turbo C ++4.5: 8
Linux GCC: 16
Visual C++: 16

Default data type of signed, unsigned, const and volatile
is int. In turbo c 3.0 size of int is two byte.
So, a = 4 and b =4
Now, a+++b
= a++ + b
= 4 + 4   //due to post increment operator.
=8

Note: In turbo c 4.5 and Linux gcc compiler size of int
is 4 byte so your out will be 16

What will be output when you will execute following c
code?

```c
#include<stdio.h>
int main(){
  signed x,a;
  unsigned y,b;
  a=(signed)10u;
  b=(unsigned)-10;
  y = (signed)10u + (unsigned)-10;
  x = y;
  printf("%d  %u\t",a,b);
  if(x==y)
    printf("%d %d",x,y);
  else if(x!=y)
    printf("%u  %u",x,y);
  return 0;
}
```

Choose all that apply:

(A) 10 -10   0 0

(B) 10 -10   65516 -10

(C) 10 -10   10 -10

(D) 10 65526   0 0

(E) Compilation error

Turbo C++ 3.0: 10 65526 0 0
Turbo C ++4.5: 10 65526 0 0

Linux GCC: 10 4294967286 0 0

Visual C++: 10 4294967286 0 0

a=(signed)10u;
signed value of 10u is +10
so, a=10
b=(unsigned)-10;
unsigned value of -10 is :
MAX_VALUE_OF_UNSIGNED_INT – 10 + 1
In turbo c 3.0 complier max value of unsigned int is 65535
So, b = 65526
y = (signed)10u + (unsigned)-10;

= 10 + 65526 = 65536 = 0 (Since 65536 is beyond the range of unsigned int. zero is its corresponding cyclic vlaue)
X = y = 0

Which of the following is integral data type?

(A) void
(B) char
(C) float
(D) double
(E) None of these

Answer: b

What will be output when you will execute following c code?

```c
#include<stdio.h>
const enum Alpha{
    X,
    Y=5,
    Z
}p=10;
int main(){
  enum Alpha a,b;
  a= X;
  b= Z;
  printf("%d",a+b-p);
  return 0;
}
```

Choose all that apply:
(A) -4
(B) -5
(C) 10
(D) 11
(E) Error: Cannot modify constant object
Turbo C++ 3.0: -4
Turbo C ++4.5: -4
Linux GCC: -4
Visual C++: -4

Default value of enum constant X is zero and
Z = Y + 1 = 5 + 1 = 6
So, a + b – p
=0 + 6 -10 = -4

What will be output when you will execute following c
code?

```c
#include<stdio.h>
int main(){
  char a=250;
  int expr;
  expr= a+ !a + ~a + ++a;
  printf("%d",expr);
  return 0;
}
```

Choose all that apply:

(A) 249

(B) 250

(C) 0

(D) -6

(E) Compilation error

Turbo C++ 3.0: -6
Turbo C ++4.5: -6
Linux GCC: -6
Visual C++: -6

```c
char a = 250;
```
250 is beyond the range of signed char. Its corresponding
cyclic value is: -6
So, a = -6
Consider on the expression:
expr= a+ !a + ~a + ++a;
Operator! , ~ and ++ have equal precedence. And it
associative is right to left.

So, First ++ operator will perform the operation. So value a will -5

Now,

Expr = -5 + !-5 + ~-5 + -5

= -5 + !-5 + 4 - 5

= -5 + 0 + 4 -5

= -6

What will be output when you will execute following c code?

```c
#include<stdio.h>
int main(){
  int a=-5;
  unsigned int b=-5u;
  if(a==b)
    printf("Avatar");
  else
    printf("Alien");
  return 0;
}
```

Choose all that apply:

(A) Avatar

(B) Alien

(C) Run time error

(D) Error: Illegal assignment

(E) Error: Don't compare signed no. with unsigned no

Turbo C++ 3.0: Avatar

Turbo C ++4.5: Avatar

Linux GCC: Avatar

Visual C++: Avatar


```c
int a=-5;
```
Here variable a is by default signed int.
```c
unsigned int b=-5u;
```
Constant -5u will convert into unsigned int. Its corresponding unsigned int value will be :

65536 – 5 + 1= 65532
So, b = 65532

In any binary operation of dissimilar data type for
example: a == b
Lower data type operand always automatically type casted
into the operand of higher data type before performing
the operation and result will be higher data type.
In c signed int is higher data type than unsigned int. So
variable b will automatically type casted into signed
int.
So corresponding signed value of 65532 is -5
Hence, a==b

What will be output when you will execute following c
code?

```c
#include<stdio.h>
enum A{
  x,y=5,
  enum B{
    p=10,q
  }varp;
}varx;

int main(){
  printf("%d %d",x,varp.q);
  return 0;
}
```

Choose all that apply:
(A) 0 11
(B) 5 10
(C) 4 11
(D) 0 10
(E) Compilation error

Answer: E Nesting enum not possible in C

What will be output if you will execute following c code?

```c
#include<stdio.h>
int main(){
   int i;
   for(i=0;i<5;i++){
      int i=10;
      printf(" %d",i);
      i++;
   }
   return 0;
}
```
(A) 10 11 12 13 14

(B) 10 10 10 10 10

(C) 0 1 2 3 4

(D) Compilation error

What will be output if you will execute following c code?

```c
#include<stdio.h>
int main(){
   int num,i=0;
   num=-++i+ ++-i;
   printf("%d",num);
   return 0;
}
```
(A) 0

(B) 1

(C) -2

(D) Compilation error

What will be output if you will execute following c code?

```c
#include<stdio.h>
int main(){
   int num,a=5;
   num=-a--+ +++a;
   printf("%d  %d",num,a);
   return 0;
}
```
(A) 1 5

(B) -1 6

(C) 1 6
(D) 0 5

What will be output if you will execute following c code?

```c
#include<stdio.h>
int main(){
    int num,a=15;
    num=- - - -a--;
    printf("%d  %d",num,a);
    return 0;
}
```
(A) 15 14
(B) 14 15
(C) 14 14
(D) 15 15

What will be output if you will execute following c code?

```c
#include<stdio.h>
int main(){
    int x,i=2;
    x=~-!++i;
    printf("%d",x);
    return 0;
}
```
(A) -2
(B) -1
(C) 0
(D) 1

What will be output if you will execute following c code?

```c
#include<stdio.h>
int main(){
    float x;
    x=0.35==3.5/10;
    printf("%f",x);
    return 0;
```

}
(A) 0.000000

(B) 1.000000

(C) 0.350000

(D) Compilation error

What will be output if you will execute following c code?

```c
#include<stdio.h>
#include<conio.h>
void main(){
  int t,a=5,b=10,c=15;
  t=(++a&&++b,++a),++a||++c;
  printf("%d  %d  %d %d",t,a,b,c);
}
```

(A) 7 8 11 15

(B) 6 7 10 14

(C) 1 8 10 15

(D) 6 18 11 15

(E) Compilation error

What will be output if you will execute following c code?

```c
#include<stdio.h>
#include<conio.h>
void main(){
  int x,y,z;
  y=(x=1,y=2,z=3);
  printf("%d  %d  %d",x,y,z);
}
```

(A) 1 2 3

(B) 1 1 3

(C) 1 3 3

(D) 1 0 3

(E) Compilation error

What will be output if you will execute following c code?

```c
#include<stdio.h>
#include<conio.h>
void main(){
  int num,a=10;
  num=a&&0+ +-1&&a;
  printf("%d  %d",num,a);
}
```

(A) 1 1

(B) 0 0

(C) 1 10

(D) 0 10

(E) Compilation error

What will be output if you will execute following c code?

```c
#include<stdio.h>
#include<conio.h>
void main(){
  int i;
  (i=8)+=1;
  printf("%d",i);
}
```

(A) 9

(B) 10

(C) 32

(D) 34

(E) Compilation error

What will be output when you will execute following c code?

```c
#include<stdio.h>
int main(){
  char a=250;
```

```c
    int expr;
    expr= a+ !a + ~a + ++a;
    printf("%d",expr);
    return 0;
}
```

Choose all that apply:

(A) 249

(B) 250

(C) 0

(D) -6

(E) Compilation error

Turbo C++ 3.0: -6
Turbo C ++4.5: -6
Linux GCC: -6
Visual C++: -6

```c
char a = 250;
```
250 is beyond the range of signed char. Its corresponding cyclic value is: -6
So, a = -6
Consider on the expression:
expr= a+ !a + ~a + ++a;
Operator! , ~ and ++ have equal precedence. And it associative is right to left.
So, First ++ operator will perform the operation. So value a will -5
Now,
Expr = -5 + !-5 + ~-5 + -5
= -5 + !-5 + 4 - 5
= -5 + 0 + 4 -5
= -6

What will be output when you will execute following c code?
```c
#include<stdio.h>
void main(){
    int check=2;
    switch(check){
```

```c
      case 1: printf("D.W.Steyn");
      case 2: printf(" M.G.Johnson");
      case 3: printf(" Mohammad Asif");
      default: printf(" M.Muralidaran");
   }
}
```

Choose all that apply:

(A) M.G.Johnson

(B) M.Muralidaran

(C) M.G.Johnson Mohammad Asif M.Muralidaran

(D) Compilation error

(E) None of the above

E x p l a n a t i o n :

If we will not use break keyword in each case the program control will come in each case after the case witch satisfy the switch condition.

What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
  int movie=1;
  switch(movie<<2+movie){
   default:printf("3 Idiots");
   case 4: printf(" Ghajini");
   case 5: printf(" Krrish");
   case 8: printf(" Race");
  }
}
```

Choose all that apply:

(A) 3 Idiots Ghajini Krrish Race

(B) Race

(C) Krrish

(D) Ghajini Krrish Race

(E) Compilation error

E x p l a n a t i o n :

We can write case statement in any order including the default case. That default case may be first case, last case or in between the any case in the switch case statement.

What will be output when you will execute following c code?

```c
#include<stdio.h>
enum actor{
  SeanPenn=5,
  AlPacino=-2,
  GaryOldman,
  EdNorton
};
void main(){
  enum actor a=0;
  switch(a){
   case SeanPenn:  printf("Kevin Spacey");
        break;
   case AlPacino:  printf("Paul Giamatti");
        break;
   case GaryOldman:printf("Donald Shuterland");
        break;
   case EdNorton:  printf("Johnny Depp");

  }
}
```

Choose all that apply:

(A) Kevin Spacey

(B) Paul Giamatti
(C) Donald Shuterland
(D) Johnny Depp
(E) Compilation error

Explanation :

Default value of enum constant
GaryOldman = -2 +1 = -1
And default value of enum constant
EdNorton = -1 + 1 = 0
Note: Case expression can be enum constant.

What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
  switch(5||2|1){
    case 3&2:printf("Anatomy of a Murder");
      break;
    case -~11:printf("Planet of Apes");
      break;
    case 6-3<<2:printf("The conversation");
      break;
  case 5>=5:printf("Shaun of the Dead");
   }
}
```

Choose all that apply:

(A) Anatomy of a Murder
(B) Planet of Apes
(C) The conversation
(D) Shaun of the Dead
(E) Compilation error

Explanation:

Consider on the expression:
5||2|1
=5|| (2|1) //Bitwise or has higher precedence
=5||3
=1

Now, value of each case expression:
3&2 = 2
-~11 = -(-12) =12
6-3<<2 = 3 <<2 = 12
5>=5 = 1

case -~11 and case 6-3<<2 have same constant expression i.e. case 12
In c duplicate case is not possible.

What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
  switch(6){
    case 6.0f:printf("Sangakkara");
       break;
    case 6.0: printf("Sehwag");
       break;
    case 6.0L:printf("Steyn");
       break;
    default:  printf("Smith");
  }
}
```

Choose all that apply:

(A) Sangakkara
(B) Sehwag
(C) Steyn
(D) Smith

(E) Compilation error

E x p l a n a t i o n :

Case expression must be integral constant expression. If it is not integer then it is automatically type casted into integer value.
so. (int)6.0f = 6
(int)6.0 = 6
(int)6.0L = 6
In c duplicate case is not possible.

What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
  switch(0X0){
    case NULL:printf("Thierry Henry");
      break;
    case '\0':printf("Steven Gerrend");
      break;
    case 0: printf("Kaka");
      break;
    default: printf("Michael Ballack");
  }

}
```

Choose all that apply:

(A) Thierry Henry
(B) Steven Gerrend
(C) Kaka
(D) Michael Ballack
(E) Compilation error

Explanation:

Macro constant NULL has be defined as 0 in stdio.h
ASCII value of character constant '\0' is 0
As we duplicate case is not possible in c.
What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
  switch(5/2*6+3.0){
    case 3:printf("David Beckham");
      break;
    case 15:printf("Ronaldinho");
      break;
    case 0:printf("Lionel Messi");
      break;
    default:printf("Ronaldo");
  }
}
```
Choose all that apply:

(A) David Beckham
(B) Ronaldinho
(C) Lionel Messi
(D) Ronaldo
(E) Compilation error

Explanation:

Consider on the expression:
5/2*6+3.0
=2*6+3.0
=12 + 3.0
=15.0

In c switch expression must return an integer value. It cannot be float, double or long double

What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
  unsigned char c=280;
  switch(c){
    printf("Start\t");
    case 280:printf("David Beckham\t");
    case 24: printf("Ronaldinho\t");
    default:  printf("Ronaldo\t");
    printf("End");
  }

}
```
Choose all that apply:

(A) Start David Beckham Ronaldinho Ronaldo End

(B) Start David Beckham Ronaldinho Ronaldo

(C) Start Ronaldinho Ronaldo End

(D) Ronaldinho Ronaldo End

(E) Compilation error

E x p l a n a t i o n :

280 is beyond the range of unsigned char. Its
corresponding cyclic value is: 24
In c switch case statement program control always move
from the case which satisfy the switch condition and end
with either break keyword, terminating} or any null
character which will come first.

13.

What will be output when you will execute following c code?

```c
#include<stdio.h>
#define TRUE 1
void main(){
  switch(TRUE){
    printf("cquestionbank.blogspot.com");
  }
}
```

Choose all that apply:

(A) cquestionbank.blogspot.com
(B) It will print nothing
(C) Runtime error
(D) Compilation error
(E) None of the above

Explanation:

In c it is possible a switch case statement without any case but it is meaning less.

What will be output when you will execute following c code?

What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
  switch(2){
    case 1L:printf("No");
    case 2L:printf("%s","I");
      goto Love;
    case 3L:printf("Please");
    case 4L:Love:printf("Hi");
```

```
  }
}
```

Choose all that apply:

(A) I

(B) IPleaseHi

(C) IHi

(D) Compilation error

(E) None of the above

E x p l a n a t i o n :

It is possible to write label of goto statement in the case of switch case statement.

What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
  int a=5;
  a=a>=4;
  switch(2){
   case 0:int a=8;
   case 1:int a=10;
   case 2:++a;
   case 3:printf("%d",a);
  }
}
```

Choose all that apply:

(A) 8

(B) 11

(C) 10

(D) Compilation error

(E) None of the above

We can not declare any variable in any case of switch case statement.

20.

What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
  int a=3,b=2;
  a=a==b==0;
  switch(1){
   a=a+10;
  }
  sizeof(a++);
  printf("%d",a);
}
```

Choose all that apply:

(A) 10

(B) 11

(C) 12

(D) 1

(E) Compilation error

E x p l a n a t i o n :

Consider on the expression:
a=a==b==0;
a=(a==b)==0; //Since associate is right to left
a =(3==2)==0
a=0==0

a=1

switch case will not affect the value of variable a.
Also sizeof operator doesn't affect the value of the any
variable