AgreeYa Technical Aptitude in C-Solved Model Question Paper

Predict the output or error(s) for the following:

1. void main()

{

int const * p=5;

printf("%d",++(*p));

}

Answer:

AgreeYa Technical Aptitude in C-Solved Model Question Paper

Compiler error: Cannot modify a constant value.

Explanation: p is a pointer to a "constant integer". But we tried to change the value of the "constant integer".

2. main()

{

char s[ ]="man";

int i;

for(i=0;s[i];i++)

printf("\n%c%c%c%c",s[i],*(s+i),*(i+s),i[s]);

AgreeYa Technical Aptitude in C-Solved Model Question Paper

```
}
```

Answer:

mmmm

aaaa

nnnn

Explanation:

s[i], *(i+s), *(s+i), i[s] are all different ways of expressing the same idea. Generally array name is the base address for that array. Here s is the base address. i is the index number/displacement from the base address. So, indirecting it with * is same as s[i].i[s] may be surprising. But in the case of C it is same as s[i].

AgreeYa Technical Aptitude in C-Solved Model Question Paper

3. main()

float me = 1.1;

double you = 1.1;

if(me==you)

printf("I love U");

else

printf("I hate U");

}

AgreeYa Technical Aptitude in C-Solved Model Question Paper

Answer:

I hate U

Explanation: For floating point numbers (float, double, long double) the values cannot be predicted exactly. Depending on the number of bytes, the precession with of the value represented varies. Float takes 4 bytes and long double takes 10 bytes. So float stores 0.9 withless precision than long double.

Rule of Thumb:

Never compare or at-least be cautious when using floating point numbers with

relational operators (== , >, <, <=, >=,!=) .

AgreeYa Technical Aptitude in C-Solved Model Question Paper

4. main()

{

static int var = 5;

printf("%d ",var--);

if(var)

main();

}

Answer:

AgreeYa Technical Aptitude in C-Solved Model Question Paper

5 4 3 2 1

Explanation:

When static storage class is given, it is initialized once. The change in the value of a static variable is retained even between the function calls. Main is also treated like any other ordinary function, which can be called recursively.

5. main()

{

int c[ ]={2.8,3.4,4,6.7,5};

int j,*p=c,*q=c;

for(j=0;j<5;j++) {

AgreeYa Technical Aptitude in C-Solved Model Question Paper

```
printf(" %d ",*c);


++q; }


for(j=0;j<5;j++){


printf(" %d ",*p);


++p; }


}
```

Answer:

2 2 2 2 2 2 3 4 6 5

Explanation: Initially pointer c is assigned to both p and q. In the first loop, since only q is incremented and not c , the value 2 will be printed 5 times. In second loop p itself is incremented. So the values 2 3 4 6 5 will be printed.

AgreeYa Technical Aptitude in C-Solved Model Question Paper

6. main()

{

extern int i;

i=20;

printf("%d",i);

}

Answer:

AgreeYa Technical Aptitude in C-Solved Model Question Paper

Linker Error : Undefined symbol '_i'

Explanation: extern storage class in the following declaration,extern int i; specifies to the compiler that the memory for i is allocated in some other program and that address will be given to the current program at the time of linking. But linker finds that no other variable of name i is available in any other program with memory space allocated for it.Hence a linker error has occurred .

7. main()

{

int i=-1,j=-1,k=0,l=2,m;

m=i++&&j++&&k++||l++;

printf("%d %d %d %d %d",i,j,k,l,m);

AgreeYa Technical Aptitude in C-Solved Model Question Paper

}

Answer:

0 0 1 3 1

Explanation : Logical operations always give a result of 1 or 0 . And also the logical AND (&&) operator has higher priority over the logical OR (||) operator. So the expression i++&& j++ && k++

is executed first. The result of this expression is 0 (-1 && -1 && 0 = 0). Now the expression is 0 || 2 which evaluates to 1 (because OR operator always gives 1 except for 0 || 0 combination- for which it gives 0). So the value of m is 1. The values of other variables are also incremented by 1.

8. main()

{

AgreeYa Technical Aptitude in C-Solved Model Question Paper

```
char *p;


printf("%d %d ",sizeof(*p),sizeof(p));


}
```

Answer:

1 2

Explanation: The sizeof() operator gives the number of bytes taken by its operand. P is a character pointer, which needs one byte for storing its value (a character). Hence sizeof(*p) gives a value of 1. Since it needs two bytes to store the address of the character pointer sizeof(p) gives 2.

9. main()

AgreeYa Technical Aptitude in C-Solved Model Question Paper

```
{

int i=3;

switch(i)

{

default:printf("zero");

case 1: printf("one");

break;

case 2:printf("two");

break;
```

AgreeYa Technical Aptitude in C-Solved Model Question Paper

```
case 3: printf("three");


break;


}


}
```

Answer :

three

Explanation :The default case can be placed anywhere inside the loop. It is executed only when all other cases doesn't match.

10. main()

AgreeYa Technical Aptitude in C-Solved Model Question Paper

```
{


printf("%x",-1<<4);


}
```

Answer:

fff0

Explanation : -1 is internally represented as all 1's. When left shifted four times the least significant 4 bits are filled with 0's.The %x format specifier specifies that the integer value be printed as a hexadecimal value.