|  |  |  |
| --- | --- | --- |
| 1. | #include<stdio.h>  void main()  {  int i=2;  printf("%d %d %d",++i,++i,++i);  } | Result :  5 4 3  Output may vary from compiler to compliler |
| 2. | int x=10,y=20,z=5,i;  i=x<y<z;  printf("\n%d",i); | 1 |
| 3. | int a=10,b,c;  (a<=20)?b=30:c=30;  printf("\nb=%d c=%d",b,c); | b=30 c=garbage |
| 4. | int a=10,b,c;  (a<=20)?b0:c=30;  printf("\nb=%d c=%d",b,c); | B=garbage c=garbage |
| 5. | Any way to write  a<=20?b=20:c=20  so that 30 is used only once | \*((a<=20)&b:&c)=30 |
| 6. | int i=1,j;  j=++i\*++i;  printf("\n%d %d",i,j); | 3 9 |
| 7. | int i=1,j;  j=++i&&++i;  printf("\n%d %d",i,j); | 3 1 |
| 8. | \*p++; | ++ has higher priority over \* so address is incremented first and then value is fetched. |
| 9. | (\*p)++ | Value is fetched first and then incremented |
| 10. | a=f1()+f2()+f3(); | Sequence of function call would be f1() then f2() and then f3() |
| 11. | int i=-3,j=2,k=0,m;  m=++i&&++j||++k;  printf("\n%d %d %d %d",i,j,k,m); | -2 3 0 1 |
| 12. | int i=-3,j=1,k=0,m;  m=++j||++i&&++k;  printf("\n%d %d %d %d",i,j,k,m); | -3 2 0 1 |
| 13. | int i=-3,j=1,k=0,m;  m=++j&&++i&&++k;  printf("\n%d %d %d %d",i,j,k,m); | -2 2 1 1 |
| 14. | float a=0.7;  if(a<0.7)  printf("C");  else  printf("C++"); | Output : C |
| 15. | float a=0.7;  if(a<0.7f)  printf("C");  else  printf("C++"); | Output : C++ |
| 16. | int sum(int);  void main()  {  printf("\n%d",sum(10));  }  int sum(int x)  {  (x>10)?return 20:return 30;  } | Return statement cannot be used with conditional operators ,so we must use it like this:  Return ((a>10)10:20); |
| 17. | #define SQR(x)(x\*x)  void main()  {  int a,b=3;  a=SQR(b+2);  printf("\n%d",a);  } | 11 |
| 18 | #define SWAP(a,b,c)(c t;t=a;a=b;b=t;)  void main()  {  int x=10,y=20;  SWAP(x,y,int);  printf("\n%d %d" ,x,y);  } | Compile time error b’coz declaration of t cannot take place inside parenthesis |
| 19. | #define SWAP(a,b,c){c t;t=a;a=b;b=t;}  void main()  {  int x=10,y=20;  SWAP(x,y,int);  printf("\n%d %d" ,x,y);  } | Success  Its limitation : Cannot swap pointers |
| 20. | How to prevent redeclaration error of header filed | #ifundef MYMACRO  #define MYMACRO  All declarrtion here  #endif |
| 21. | void main()  {  #ifdef NOTE  /\*Unterminated comment  int a;  a=10;  #else  int a;  a=20;  #endif  printf("\n%d",a);  } | Comment will cause everything commented |
| 22. | #define MESS junk  void main()  {  printf("MESS");  } | MESS |
| 23. | #define MESS “junk”  void main()  {  printf(MESS);  } | Junk |
| 24. | #define print(int) printf("int=%d",int)  void main()  {  int x=2,y=3,z=4;  print(x);  print(y);  print(z);  } | Int=2int=3int=4  #define print(int) printf(#int"=%d",int)  void main()  {  int x=2,y=3,z=4;  print(x);  print(y);  print(z);  }  This will print x=2y=3z=4  If parameter name is preceded by # in macro expansion the combination of # and parameter will be expanded into a quoted string with the parameter replaced by actual arguments. This can be combined with string concatenation to print the output desired in our program. |
| 25. | Static int i=10 ; this declaration is in first.h  Now we include it in second.h and both first & second.h are included in main.c then we get redeclaration error. To avoid this we use #pragma once just about static int i=10; |  |