Pre Processor Practice sheet

**1. #define call(x) #x**  
**void main(){**  
**printf("%s",call(c/c++));**  
**}**  
**2. #define p(a,b) a##b**  
**#define call(x) #x**  
**int main(){**  
**do{**  
**int i=15,j=3;**  
**printf("%d",p(i-+,+j));**  
**}**  
**while(\*(call(625)+3));**  
**}**  
**3. #define PRINT printf("Star Wars");printf(" Psycho");**  
**#include**  
**void main(){**  
**int x=1;**  
**if(x--)**  
**PRINT**  
**else**  
**printf("The Shawshank Redemption");**  
**}**  
**4. What is the output of this C code?**  
**#include**  
**#define foo(m, n) m \* n = 10**  
**int main()**  
**{**  
**printf("in main\n");**  
**}**  
**a) In main**  
**b) Compilation error as lvalue is required for the expression**  
**m\*n=10**  
**c) Preprocessor error as lvalue is required for the expression**  
**m\*n=10**  
**d) None of the mentioned**  
**5. C preprocessor is conceptually the first step during compilation**  
**a) true**  
**b) false**  
**c) Depends on the compiler**  
**d) Depends on the standard**  
**6. Preprocessor feature that supply line numbers and filenames to**  
**compiler is called?**  
**a) Selective inclusion**  
**b) macro substitution**  
**c) Concatenation**  
**d) Line control**  
**7. #include are \_\_\_\_\_\_\_ files and #include “somefile.h” \_\_\_\_\_\_\_\_**  
**files.**  
**a) Library, Library**  
**b) Library, user-created header**  
**c) User-created header, library**  
**d) They can include all types of file**  
**8. A preprocessor is a program**  
**a) That processes its input data to produce output that is used as**  
**input to another program**  
**b) That is nothing but a loader**  
**c) That links various source files**  
**d) All of the mentioned**  
**9. Which of the following are C preprocessors?**  
**a) #ifdef**  
**b) #define**  
**c) #endif**  
**d) All of the mentioned**  
**10.#include statement must be written**  
**a) Before main()**  
**b) Before any scanf/printf**  
**c) After main()**  
**d) It can be written anywhere**  
**11.#pragma exit is primarily used for?**  
**a) Checking memory leaks after exitting the program**  
**b) Informing Operating System that program has terminated**  
**c) Running a function at exitting the program**  
**d) No such preprocessor exist**  
**12.What is the output of this C code?**  
**#include**  
**int main()**  
**{**  
**int one = 1, two = 2;**  
**#ifdef next**  
**one = 2;**  
**two = 1;**  
**#endif**  
**printf("%d, %d", one, two);**  
**}**  
**a) 1, 1**  
**b) 1, 2**  
**c) 2, 1**  
**d) 2, 2**  
**13. The C-preprocessors are specified with \_\_\_\_\_\_\_\_\_symbol.**  
**a) #**  
**b) $**  
**c) ” ”**  
**d) None of the mentioned**  
**14. The #include directive**  
**a) Tells the preprocessor to grab the text of a file and place it**  
**directly into the current file**  
**b) Statements are typically placed at the top of a program**  
**c) both a & b**  
**d) None of a & b**  
**15. The preprocessor provides the ability for \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.**  
**a) The inclusion of header files**  
**b) The inclusion of macro expansions**  
**c) Conditional compilation and line control.**  
**d) All of the mentioned**  
**16.If #include is used with file name in angular brackets**  
**a) The file is searched for in the standard compiler include paths**  
**b) The search path is expanded to include the current source**  
**directory**  
**c) Both a & b**  
**d) None of the mentioned**  
**17.What is the sequence for preprocessor to look for the file within**  
**<> ?**  
**a) The predefined location then the current directory**  
**b) The current directory then the predefined location**  
**c) The predefined location only**  
**d) The current directory location**  
**18. What would happen if you create a file stdio.h and use #include**  
**“stdio.h” ?**  
**a) The predefined library file will be selected**  
**b) The user-defined library file will be selected**  
**c) Both the files will be included**  
**d) The compiler won’t accept the program**  
**19.Can function definition be present in header files?**  
**a) Yes**  
**b) No**  
**c) Depends on the compiler**  
**d) Depends on the standard**  
**20.Comment on the output of this C code?**  
**#include**  
**#include "test.h"**  
**#include "test.h"**  
**int main()**  
**{**  
**//some code**  
**}**  
**a) true**  
**b) Compile time error**  
**c) false**  
**d) Depends on the compiler  
<st< strong="" style="box-sizing: border-box;">ro</st<>**ng> **21.What is the output of this C code?**  
**#include**  
**#define foo(m, n) m ## n**  
**void myfunc();**  
**int main()**  
**{**  
**myfunc();**  
**}**  
**void myfunc()**  
**{**  
**printf("%d\n", foo(2, 3));**  
**}**  
**a) 23**  
**b) 2 3**  
**c) Compile time error**  
**d) Undefined behaviour**  
**22.If the file name is enclosed in double quotation marks**  
**a) The preprocessor treats it as a user-defined file**  
**b) The preprocessor treats it as a system-defined file**  
**c) Both a & b**  
**d) None of the mentioned**  
**23.If the file name is enclosed in angle brackets**  
**a) The preprocessor treats it as a user-defined file**  
**b) The preprocessor treats it as a system-defined file**  
**c) Both a & b**  
**d) None of the mentioned**  
**24.What is the output of this C code?**  
**#include (stdio.h)**  
**void main()**  
**{**  
**printf("hello");**  
**}**  
**a) hello**  
**b) Nothing**  
**c) compile time error**  
**d) Depends on compiler**  
**25.The below two lines are equivalent to**  
**#define C\_IO\_HEADER**  
**#include C\_IO\_HEADER**  
**a) #include**  
**b) #include”printf”**  
**c) #include”C\_IO\_HEADER”**  
**d) #include**  
**26.What is the output of this C code?**  
**#include**  
**#include "printf"**  
**void main()**  
**{**  
**printf("hello");**  
**}**  
**a) hello**  
**b) Error**  
**c) Depends on compiler**  
**d) Varies**  
**27.Which of the following file extensions are accepted with #include?**  
**a) .h**  
**b) .in**  
**c) .com**  
**d) All of the mentioned**  
**28.Which of the following names for files not accepted?**  
**a) header.h.h**  
**b) 123header.h**  
**c) \_head\_er.h**  
**d) None of the mentioned**  
**29.What is the output of this C code?**  
**#include**  
**#define SYSTEM 20**  
**int main()**  
**{**  
**int a = 20;**  
**#if SYSTEM == a**  
**printf("HELLO ");**  
**#endif**  
**#if SYSTEM == 20**  
**printf("WORLD\n");**  
**#endif**  
**}**  
**a) HELLO**  
**b) WORLD**  
**c) HELLO WORLD**  
**d) No Output**  
**30.Comment on the following code?**  
**#include**  
**#define Cprog**  
**int main()**  
**{**  
**int a = 2;**  
**#ifdef Cprog**  
**a = 1;**  
**printf("%d", Cprog);**  
**}**  
**a) No output on execution**  
**b) Output as 1**  
**c) Output as 2**  
**d) Compile time error**  
**31.The “else if” in conditional inclusion is written by?**  
**a) #else if**  
**b) #elseif**  
**c) #elsif**  
**d) #elif**  
**32.What is the output of this C code?**  
**#include**  
**#define COLD**  
**int main()**  
**{**  
**#ifdef COLD**  
**printf("COLD\t");**  
**#undef COLD**  
**#endif**  
**#ifdef COLD**  
**printf("HOT\t");**  
**#endif**  
**}**  
**a) HOT**  
**b) COLD**  
**c) COLD HOT**  
**d) No Output**  
**33.Which of the following sequences are unaccepted in C language?**  
**a) #if**  
**#else**  
**#endif**  
**b) #if**  
**#elif**  
**#endif**  
**c) #if**  
**#if**  
**#endif**  
**d) #if**  
**#undef**  
**#endif**  
**34.In a conditional inclusion, if the condition that comes after the**  
**if holds.**  
**a) Then the code up to the following #else or #elif or #endif is**  
**compiled**  
**b) Then the code up to the following #endif is compiled even if #else or #elif is present**  
**c) Both a & b**  
**d) None of the mentioned**  
**35.Conditional inclusion can be used for**  
**a) Preventing multiple declarations of a variable**  
**b) Check for existence of a variable and doing something if it**  
**exists**  
**c) Preventing multiple declarations of same function**  
**d) All of the mentioned**  
**36.The #elif directive cannot appear after the preprocessor #else**  
**directive.**  
**a) true**  
**b) false**  
**c) None of the mentioned**  
**d) Varies**  
**37. What is the output of this C code?**  
**#include**  
**void main()**  
**{**  
**#define max 37;**  
**printf("%d", max);**  
**}**  
**a) 37**  
**b) Compile time error**  
**c) Varies**  
**d) Depends on compiler**  
**38.What is the output of this C code?**  
**#include**  
**void main()**  
**{**  
**#define max 37**  
**printf("%d", max);**  
**}**  
**a) 37**  
**b) Run time error**  
**c) Varies**  
**d) Depends on compiler**  
**39.What is the output of this C code?**  
**#include**  
**void main()**  
**{**  
**#define const int**  
**const max = 32;**  
**printf("%d", max);**  
**}**  
**a) Run time error**  
**b) 32**  
**c) int**  
**d) const**  
**40.What is the output of this C code?**  
**#include**  
**void main()**  
**{**  
**#define max 45**  
**max = 32;**  
**printf("%d", max);**  
**}**  
**a) 32**  
**b) 45**  
**c) Compile time error**  
**d) Varies**  
**41.What is the output of this C code?**  
**#include**  
**# define max**  
**void m()**  
**{**  
**printf("hi");**  
**}**  
**void main()**  
**{**  
**max;**  
**m();**  
**}**  
**a) Run time error**  
**b) hi hi**  
**c) Nothing**  
**d) hi**  
**42.What is the output of this C code?**  
**#include**  
**#define A 1 + 2**  
**#define B 3 + 4**  
**int main()**  
**{**  
**int var = A \* B;**  
**printf("%d\n", var);**  
**}**  
**a) 9**  
**b) 11**  
**c) 12**  
**d) 21**  
**43.Which of the following Macro substitution are accepted in C?**  
**a) #define A #define**  
**A VAR 20**  
**b) #define A define**  
**#A VAR 20**  
**c) #define #A #define**  
**#A VAR 20**  
**d) None of the mentioned**  
**44.Comment on the following code?**  
**#include**  
**#define var 20);**  
**int main()**  
**{**  
**printf("%d\n", var**  
**}**  
**a) No errors, it will show the output 20**  
**b) Compile time error, the printf braces aren’t closed**  
**c) Compile time error, there are no open braces in #define**  
**d) Both ( b) and (c).**  
**45.Which of the following properties of #define not true?**  
**a) You can use a pointer to #define**  
**b) #define can be made externally available**  
**c) They obey scope rules**  
**d) All of the mentioned**  
**46.What is the output of this C code?**  
**#include**  
**#define SYSTEM 20**  
**int main()**  
**{**  
**int a = 20;**  
**#if SYSTEM == a**  
**printf("HELLO ");**  
**#endif**  
**#if SYSTEM == 20**  
**printf("WORLD\n");**  
**#endif**  
**}**  
**a) HELLO**  
**b) WORLD**  
**c) HELLO WORLD**  
**d) No Output**  
**47.Comment on the following code?**  
**#include**  
**#define Cprog**  
**int main()**  
**{**  
**int a = 2;**  
**#ifdef Cprog**  
**a = 1;**  
**printf("%d", Cprog);**  
**}**  
**a) No output on execution**  
**b) Output as 1**  
**c) Output as 2**  
**d) Compile time error**  
**48.The “else if” in conditional inclusion is written by?**  
**a) #else if**  
**b) #elseif**  
**c) #elsif**  
**d) #elif**  
**49.What is the output of this C code?**  
**#include**  
**#define COLD**  
**int main()**  
**{**  
**#ifdef COLD**  
**printf("COLD\t");**  
**#undef COLD**  
**#endif**  
**#ifdef COLD**  
**printf("HOT\t");**  
**#endif**  
**}**  
**a) HOT**  
**b) COLD**  
**c) COLD HOT**  
**d) No Output**  
**50.Which of the following sequences are unaccepted in C language?**  
**a) #if**  
**#else**  
**#endif**  
**b) #if**  
**#elif**  
**#endif**  
**c) #if**  
**#if**  
**#endif**  
**d) #if**  
**#undef**  
**#endif**  
**51.In a conditional inclusion, if the condition that comes after the if holds.**  
**a) Then the code up to the following #else or #elif or #endif is**  
**compiled**  
**b) Then the code up to the following #endif is compiled even if**  
**#else or #elif is present**  
**c) Both a & b**  
**d) None of the mentioned**  
**52.Conditional inclusion can be used for**  
**a) Preventing multiple declarations of a variable**  
**b) Check for existence of a variable and doing something if it**  
**exists**  
**c) Preventing multiple declarations of same function**  
**d) All of the mentioned**  
**53.The #elif directive cannot appear after the preprocessor #else**  
**directive.**  
**a) true**  
**b) false**  
**c) None of the mentioned**  
**d) Varies**  
**54.For each #if, #ifdef, and #ifndef directive.**  
**a) There are zero or more #elif directives**  
**b) Zero or one #else directive**  
**c) One matching #endif directive**  
**d) All of the mentioned**  
**55.The #else directive is used for**  
**a) Conditionally include source text if the previous #if, #ifdef,**  
**#ifndef, or #elif test fails.**  
**b) Conditionally include source text if a macro name is not defined**  
**c) Conditionally include source text if a macro name is defined**  
**d) Ending conditional text**  
**56.What is the output of this C code?**  
**#include**  
**#define MIN 0**  
**#if MIN**  
**#define MAX 10**  
**#endif**  
**int main()**  
**{**  
**printf("%d %d\n", MAX, MIN);**  
  
**}**  
**a) 10 0**  
**b) Compile time error**  
**c) Undefined behaviour**  
**d) None of the mentioned**  
**57.What is the output of this C code?**  
**#include**  
**#define MIN 0**  
**#ifdef MIN**  
**#define MAX 10**  
**#endif**  
**int main()**  
**{**  
**printf("%d %d\n", MAX, MIN);**  
  
**}**  
**a) 10 0**  
**b) Compile time error**  
**c) Undefined behaviour**  
**d) None of the mentioned**  
**58. What is the output of this C code?**  
**#include**  
**#define MIN 0**  
**#if defined(MIN) + defined(MAX)**  
**#define MAX 10**  
**#endif**  
**int main()**  
**{**  
**printf("%d %d\n", MAX, MIN);**  
  
**}**  
**a) 10 0**  
**b) Compile time error**  
**c) Undefined behaviour**  
**d) Somegarbagevalue 0**  
**59.What is the output of this C code?**  
**#include**  
**#define MIN 0**  
**#if defined(MIN) - (!defined(MAX))**  
**#define MAX 10**  
**#endif**  
**int main()**  
**{**  
**printf("%d %d\n", MAX, MIN);**  
  
**}**  
**a) 10 0**  
**b) Compile time error**  
**c) Undefined behaviour**  
**d) Somegarbagevalue 0**  
**60. What is the output of this C code?**  
**#include**  
**#define MIN 0**  
**#ifdef(MIN)**  
**#define MAX 10**  
**#endif**  
**int main()**  
**{**  
**printf("%d %d\n", MAX, MIN);**  
  
**}**  
**a) 10 0**  
**b) Compile time error**  
**c) Both b and c**  
**d) Preprocessor error**  
**61. What is the output of code given below?**  
**#include**  
**#define MIN 0);**  
**#ifdef MIN**  
**#define MAX 10**  
**#endif**  
**int main()**  
**{**  
**printf("%d %d\n", MAX, MIN**  
  
**}**  
**a) 10 0**  
**b) Compile time error due to illegal syntax for printf**  
**c) Undefined behaviour**  
**d) Compile time error due to illegal MIN value**  
**62. #define max**  
**int main(){**  
**printf("%d",max);**  
**}**  
**63. int main(){**  
**int a=0;**  
**#if (a==0)**  
**printf("Equal");**  
**#else if**  
**printf("Not equal");**  
**#endif**  
**}**  
**64. int main(){**  
**#ifndef NULL**  
**#define NULL 5**  
**#endif**  
**printf("%d",NULL+sizeof(NULL));**  
**}**  
**65. Property which allows to produce different executable for different**  
**platforms in C is called?**  
**a) File inclusion**  
**b) Selective inclusion**  
**c) Conditional compilation**  
**d) Recursive macros**  
**66. #include is called**  
**a) Preprocessor directive**  
**b) Inclusion directive**  
**c) File inclusion directive**  
**d) None of the mentioned**  
**67. C preprocessors can have compiler specific features.**  
**a) true**  
**b) false**  
**c) Depends on the standard**  
**d) Depends on the platform**  
**68. #include**  
**void main(){**  
**printf("\nS1");**  
**#if 5!=5**  
**printf("\n");**  
**printf("\nS2");**  
**#endif**  
**printf("\nS2");**  
**}**  
**69. void main(){**  
**printf("\nS1");**  
**#if 5>8!=0**  
**printf("\n");**  
**printf("\nS2");**  
**#else**  
**printf("\nS3");**  
**printf("\Ns4");**  
**#endif**  
**printf("\Ns5");**  
**}**  
**70. void main(){**  
**printf("S1");**  
**#if !5**  
**printf("S2");**  
**printf("S3");**  
**#elsif 5>2**  
**printf("\ns4");**  
**#else**  
**printf("\nS5");**  
**#endif**  
**}**  
**71.void main(){**  
**printf("\nS1");**  
**#ifdef TEST**  
**printf("\nS2");**  
**printf("\n");**  
**#endif**  
**printf("\nS3");**  
**}**  
**72.#include**  
**#define TEST**  
**void main() {**  
**#ifndef TEST**  
**#error TEST SHOULD BE DEFINED**  
**#endif#ifdef TEST**  
**printf("\nS1");**  
**printf("\nHello");**  
**#endif**  
**}**  
**73. #define TEST**  
**void main() {**  
**#ifndef TEST**  
**#error TEST MUST BE DEFINED**  
**#endif**  
**#ifdef TEST**  
**printf("\nS1");**  
**printf("\nS2");**  
**#endif**  
**}**  
**74. #define SQR(a) a\*a**  
**void main(){**  
**int x,y;**  
**x=2;**  
**y=SQR(x);**  
**printf("\nSQR of %d is: %d",x,y);**  
**}**  
**75. #pragma warn -par**  
**#pragma warn -rvl**  
**#pragma warn -rch**  
**int abc(int a){**  
**printf("\nHello abc");**  
**}**  
**int main(){**  
**abc(10);**  
  
**getch();**  
**}**  
**76. int \_pascal ABC(int x,int y){**  
**printf("\nx=%d y=%d",x,y);**  
**}**  
**void main(){**  
**int a;**  
**a=5;**  
**ABC(++a,a++);**  
**ABC(a++,++a);**  
**printf("\na=%d",a);**  
**}**  
**77.#define START main() {**  
**#define PRINT printf("\*\*\*\*\*\*\*");**  
**#define END }**  
**START**  
**PRINT**  
**END**  
**78.#define SQR(x) x\*x**  
**void main(){**  
**int a,b,c;**  
**a=SQR(2+3);**  
**printf("\nSQR of 2+3=%d",a);**  
**b=3;**  
**c=4;**  
**a=SQR(b+c);**  
**printf("\nSQR of %d+%d=%d",b,c,a);**  
**}**  
**79.#define SQR(x) (x)\*(x)**  
**#define CUBE(x) SQR(x) \*(X)**  
**void main(){**  
**int a;**  
**a=CUBE(2);**  
**printf("\n%d",a);**  
**}**  
**80.#define A 2+3**  
**#define B 3+4**  
**void main(){**  
**int x;**  
**x=A\*B;**  
**printf("%d",x);**  
**}**  
**81.void abc();**  
**void xyz();**  
**#pragma startup abc**  
**#pragma exit xyz**  
**void abc(){**  
**printf("\nFrom abc:");**  
**}**  
**void main(){**  
**printf("\nFrom Main:");**  
**}**  
**void xyz(){**  
**printf("\nFrom xyz");**  
**}**  
**82.#define A 10**  
**void main(){**  
**int i;**  
**i=A;**  
**printf("\n%d %d",i,A);**  
**}**  
**83.#define A 2+3**  
**#define B 4+5**  
**void main(){**  
**float r;**  
**r=A/B;**  
**printf("%f",r);**  
**}**  
**84.#include**  
**#define A 1**  
**void main(){**  
**printf("\na=%d",A);**  
**#undef A**  
**#define A 10**  
**printf("\na=%d",A);**  
**#undef A**  
**#define A 20**  
**printf("\na=%d",A);**  
**}**  
**85.#define MAX 5;**  
**void main()**  
**{**  
**printf("%d",MAX);**  
**}**  
**86.#define MSSG printf("S1\n");**  
**main()**  
**{**  
**MSSG**  
**}**  
**87.#define PROD(x,y) ((x)\*(y))**  
**void main()**  
**{**  
**int a=3,b=4;**  
**printf("a and b=%d",PROD(a,b));**  
**}**  
**88.#define A 50**  
**#define B A+100**  
**void main()**  
**{**  
**int i,j;**  
**i=B/20;**  
**j=500-B;**  
**printf("i =%d,j=%d\n",i,j);**  
**}**  
**89.#define NEW\_LINE printf("\n");**  
**#define BLANK\_LINES(n) (inti;for(i=0;i<="" strong="" style="box-sizing: border-box;">  
void main()  
{  
printf("S1");  
NEW\_LINE  
printf("S2");  
BLANK\_LINES(3)  
printf("S3");  
NEW\_LINE  
}  
90.#define INFINITE while(1)  
#define CHECK(a) if(a==0) break  
main()  
{  
int x=2;  
INFINITE  
{  
printf("%d",x--);  
}  
}  
91.#define ABS(x) ((x)<0 ?-(x):(x))  
void main()  
{  
int array[4]={1,-2,3,-4};  
int \*p=arra+3;  
while(p>=array)  
{  
printf("%d ",ABS(\*p));  
p--;  
}  
}  
92.#define . ;  
void main()  
{  
printf("S1").  
printf(" S2").  
}  
93.#define CUBE(x) (x\*x\*x)  
void main()  
{  
printf("%d\n",CUBE(1+2));  
}  
94.#define CUBE(x) ((x)\*(x)\*(x))  
void main()  
{  
int i=1;  
while(i<=8)  
printf("%d\n",CUBE(i++));  
}  
95.#define SWAP(dtype,x,y) {dtype t; t=x+y,x=t-x,y=t-y}  
void main()  
{  
int a=1,b=2,x=3,y=4,s=25,t=26;  
SWAP(int,a,b)  
SWAP(int,x,y)  
SWAP(int,s,t)  
printf("a=%d,b=%d,x=%d,y=%d,s=%d,t=%d\n",a,b,x,y,s,t);  
}  
96.#define INC(dtype,x,i) x=x+i  
void main()  
{  
int arr[5]={20,34,56,12,96},\*ptr=arr;  
INC(int,arr[2],3);  
INC(int\*,ptr,2);  
printf("\*ptr=%d\n",\*ptr);  
}  
97.#define INT int  
{  
INT a=2,\*p=&a;  
printf("%d %d\n",a,\*p);  
}  
98.#define Y 10  
void main()  
{  
#if X||Y&&Z  
printf("S1");  
#else  
printf("S2");  
#endif  
}  
99.void main()  
{  
int x=3,y=4,z;  
z=x+y;  
#include  
printf("%d\n",z);  
}  
100.#define DIFF(FNAME,DTYPE,RTYPE) \  
RTYPE FNAME (DTYPE X,DTYPE Y){return X-Y}  
DIFF(diff\_int,int,int)  
DIFF(diff\_iptr,int\*,int)  
DIFF(diff\_float,float,float);  
DIFF(diff\_fptr,float\*,int);  
void main()  
{  
int iarr[5]={1,2,3,4,5},a,p,q;  
float farr[7]={1.2,2.3,3.4,4.5,5.6,6.7,7.8},b;  
a=diff\_int(iarr[4],iarr[1]);  
b=diff\_float(farr[6],farr[2]);  
p=diff\_iptr(&iarr[4],&iarr[1]);  
printf("a=%d,b=%.1f,p=%d,q=%d\n",a,b,p,q);  
}  
101.#define MAX 3  
void main()  
{  
printf("S1 %d\n",MAX);  
#undef MAX  
#ifdef MAX  
printf("Have a good day");  
#endif  
}  
102.#define PRINT1(message) printf(message);  
#define PRINT2(message) printf("message");  
#define PRINT3(message) printf(#message);  
main()  
{  
PRINT1("S1");  
PRINT2("S2");  
PRINT3("S3");  
}  
103.#define show(value) printf(#value " = %d\n",value);  
void main()  
{  
int a=10,b=5,c=4;  
show(a/b\*c);  
}  
104.#define MACRO(a) if(a<=5) printf(#a"=%d\n",a);  
void main()  
{  
int x=6,y=15;  
if(x<=y)  
MACRO(x);  
else  
MACRO(y);  
}  
105.void main()  
{  
#line 100 "system.c"  
printf("%d %s\n",\_\_LINE\_\_,\_\_FILE\_\_);  
}  
106.For which of the following, “PI++;” code will fail?  
a) #define PI 3.14  
b) char \*PI = “A”;  
c) float PI = 3.14;  
d) Both (A) and (B)  
107. What is the output of this C code?  
enum birds {SPARROW, PEACOCK, PARROT};  
enum animals {TIGER = 8, LION, RABBIT, ZEBRA};  
int main()  
{  
enum birds m = TIGER;  
int k;  
k = m;  
printf("%d\n", k);  
  
}  
a) 0  
b) Compile time error  
c) 1  
d) 8  
108. What is the output of this C code?  
#define a 10  
int main()  
{  
const int a = 5;  
printf("a = %d\n", a);  
}  
a) a = 5  
b) a = 10  
c) Compilation error  
d) Runtime error  
109. which is incorrect about preprocessors  
a. Each preprocessor directive starts with a # symbol.  
b. There can be only one directive on a line.  
c. There is no semicolon at the end of a directive.  
d. To continue a directive on next line, we should place a backslash at the end of the line.  
e. The preprocessor directives can be placed anywhere in a program (inside or outside functions) but they are usually written at the beginning of a program.  
f. A directive is active from the point of its appearane till the end of the program.  
g. A directive is active within a function which has defined it.  
110. The main functions performed by the preprocessor directives are.  
a. Simple Macro Subtitution  
b. Macros with arguments  
c. Conditional Compilation  
d. Including files  
e. Error generations, pragmas and predefined macro names.  
f. conditiona execution**