1)

typedefint (\*test) ( float \* , float\*)

testtmp;

type of tmp is   
  
(a) Pointer to function of having two arguments that is pointer to float  
(b) int  
(c) Pointer to function having two argument that is pointer to float and return int  
(d) None of the above

ans (c)

|  |  |
| --- | --- |
| 2). | Declare the following statement? "A pointer to an array of three chars". |
| |  |  |  |  | | --- | --- | --- | --- | | [A.](javascript:%20void%200;) | char \*ptr[3](); | [B.](javascript:%20void%200;) | char (\*ptr)\*[3]; | | [C.](javascript:%20void%200;) | char (\*ptr[3]); | [D.](javascript:%20void%200;) | char (\*ptr)[3]; | |

Answer (d)

3)void f1(int \*, int);

void f2(int \*, int);

void(\*p[2]) ( int \*, int);

main()

{

int a;

int b;

p[0] = f1;

p[1] = f2;

a=3;

b=5;

p[0](&a , b);

printf("%d\t %d\t" , a ,b);

p[1](&a , b);

printf("%d\t %d\t" , a ,b);

}

void f1( int\* p , int q)

{

inttmp;

tmp =\*p;

\*p = q;

q= tmp;

}

void f2( int\* p , int q)

{

inttmp;

tmp =\*p;

\*p = q;

q= tmp;

}

The output for this program is:   
  
(a) 5 5 5 5  
(b) function declaration mismatch  
(c) 5 3 5 3  
(d) 3 3 3 3

Ans (a)

4)

#include<stdarg.h>

int ripple ( int , ...);

main()

{

int num;

num = ripple ( 3, 5,7);

printf( " %d" , num);

}

int ripple (int n, ...)

{

int i , j;

int k;

va\_list p;

k= 0;

j = 1;

va\_start( p , n);

for (; j<n; ++j)

{

i = va\_arg( p , int);

for (; i; i &=i-1 )

++k;

}

return k;

}

The output for this program is:   
  
(a) 7  
(b) 6  
(c) 5  
(d) 3

The answer is (c)

|  |
| --- |
| 5)  The value of j at the end of the execution of the following C program.  intincr (int i)  {     static int count = 0;     count = count + i;     return (count);  }  main ()  {     inti,j;     for (i = 0; i <=4; i++)        j = incr(i);  } |

1. 10  
   (b) 4  
   (c) 6  
   (d) 7

Ans: a

6)

main()

{

struct node

{

int a;

int b;

int c;

};

struct node s= { 3, 5,6 };

struct node \*pt = &s;

printf("%d" , \*(int\*)pt);

}

The output for this program is:   
(a) 3  
(b) 5  
(c) 6  
(d) 7

Answer (a)

7)

Consider the following C function definition:  
int Trial (int a, int b, int c)   
{   
   if ((a > = b) && (c < b)) return b;   
   else if (a > = b) return Trial (a,c,b);   
   else return Trial (b,a,c);   
}  
The function Trial:    
(a) Finds the maximum of a, b, and c   
(b) Finds the minimum of a, b and c   
(c) Finds the middle number of a, b, c   
(d) None of the above    
  
Ans: (c)

8)

|  |
| --- |
| #include <stdio.h>    void myStrcat(char \*a, char \*b)  {      int m = strlen(a);      int n = strlen(b);      int i;      for (i = 0; i <= n; i++)         a[m+i]  = b[i];  }    int main()  {      char \*str1 = "Geeks ";      char \*str2 = "Quiz";      myStrcat(str1, str2);      printf("%s ", str1);      return 0;  } |

Which of the following changes can correct the program so that it prints “Geeks Quiz”?  
**(A)** char \*str1 = “Geeks “; can be changed to char str1[100] = “Geeks “;  
**(B)** char \*str1 = “Geeks “; can be changed to char str1[100] = “Geeks “; and a line a[m+n-1] = ‘\0′ is added at the end of myStrcat  
**(C)** A line a[m+n-1] = ‘\0′ is added at the end of myStrcat  
**(D)** A line ‘a = (char \*)malloc(sizeof(char)\*(strlen(a) + strlen(b) + 1)) is added at the beginning of myStrcat()  
  
  
**Answer:** **(A)**

**9)**

What is the output of following program?

# include <stdio.h>

int main()

{

char str1[] = "GeeksQuiz";

char str2[] = {'G', 'e', 'e', 'k', 's', 'Q', 'u', 'i', 'z'};

int n1 = sizeof(str1)/sizeof(str1[0]);

int n2 = sizeof(str2)/sizeof(str2[0]);

printf("n1 = %d, n2 = %d", n1, n2);

return 0;

}

**(A)**n1 = 10, n2 = 9

**(B)**n1 = 10, n2 = 10

**(C)**n1 = 9, n2 = 9

**(D)**n1 = 9, n2 = 10

**Answer:(A)**

10)

main()

{

int a[5] = {1,2,3,4,5};

int \*ptr = (int\*)(&a+1);

printf("%d %d" , \*(a+1), \*(ptr-1) );

}

The output for this program is:   
  
(a) 2 2  
(b) 2 1  
(c) 2 5  
(d) None of the above

Answer (c)

11)

|  |
| --- |
| #include<stdio.h>  void swap(char \*str1, char \*str2)  {    char \*temp = str1;    str1 = str2;    str2 = temp;  }    int main()  {    char \*str1 = "Geeks";    char \*str2 = "Quiz";    swap(str1, str2);    printf("str1 is %s, str2 is %s", str1, str2);    return 0;  } |

**(A)**str1 is Quiz, str2 is Geeks  
**(B)** str1 is Geeks, str2 is Quiz  
**(C)** str1 is Geeks, str2 is Geeks  
**(D)** str1 is Quiz, str2 is Quiz  
  
  
**Answer:(B)**

*12)*

*#include<stdio.h>*

*main()*

*{*

*struct x*

*{*

*int x;*

*struct y*

*{*

*int x;*

*};*

*struct y \*q;*

*};*

*}*

a)no error

b)error-Redeclaration of x

c)error-Declaration terminated incorrectly

d)error-Nesting of structures not allowed

ans: c

13) What value will be stored in z if the following code is executed?

main()

{

int x=5,y=-10,z;

a=4,b=2;

z=x+++++y\*b/a;

}

1. -2 (b) 0 (c) 1 (d) 2

Ans: c

14)

main()

{

int a, b,c, d;

a=3;

b=5;

c=a,b;

d=(a,b);

printf("c=%d" ,c);

printf("d=%d" ,d);

}

The output for this program is:  
  
(a) c=3 d=3  
(b) c=5 d=3  
(c) c=3 d=5  
(d) c=5 d=5

Answer (c)

15) What will be the output of the following code?

int a=1; b=2; c=3; \*pointer;

pointer=&c;

a=c/\*pointer;

b=c;

printf(“a=%d b=%d”,a, b);

1. a=1 b=3 (b) a=3 b=3 (c) 3 2 (d) Error

ans: d

16) main ( )

{

static char \*s[ ] = {“black”, “white”, “yellow”, “violet”};

char \*\*ptr[ ] = {s+3, s+2, s+1, s}, \*\*\*p;

p = ptr;

\*\*++p;

printf(“%s”,\*--\*++p + 3);

}

a)te b)black c)low d)ck

ans: d

17) What is the size of ‘q’ in the following program?

Assume int takes 4 bytes.

union

{

int x;

char y;

struct

{

char x;

char y;

int xy;

}p;

}q;

1. 11 (b) 6 (c) 4 (d) 5

Ans: b

18)

main()

{

int i=3;

int j;

j = sizeof(++i+ ++i);

printf("i=%d j=%d", i ,j);

}

The output for this program is:  
  
(a) i=4 j=2  
(b) i=3 j=2  
(c) i=3 j=4  
(d) i=3 j=6

Ans (b)

19) What is the size of the ptr1 and ptr2?

struct x

{

int j;

char k[100];

unsigned i;

};

int \*ptr1;

struct x \*ptr2;

1. same depending on the model used (b) 2, 104 (c) 2, undefined for memory is not allocated (d) 2,4

ans: a

20)

main()

{

int i;

i = abc();

printf("%d",i);

}

abc()

{

\_AX = 1000;

}

a)garbage value b)0 c)error d)1000

ans: d