1)

void foo(int [][3] );

main()

{

int a [3][3]= { { 1,2,3} , { 4,5,6},{7,8,9}};

foo(a);

printf("%d" , a[2][1]);

}

void foo( int b[][3])

{

++ b;

b[1][1] =9;

}

The output for this program is:  
  
(a) 8  
(b) 9  
(c) 7  
(d) None of the above

Answer (b)

*2)*

|  |
| --- |
| main()  {              int i=0;              while(+(+i--)!=0)                          i-=i++;              printf("%d",i);  }  a)-1 b)1 c)0 d)infinite loop |

**Answer:** -1

3)

int \*f1(void)

{

int x =10;

return(&x);

}

int \*f2(void)

{

Int\*ptr;

\*ptr =10;

return ptr;

}

int \*f3(void)

{

int \*ptr;

ptr=(int\*) malloc(sizeof(int));

return ptr;

}

Which of the above three functions are likely to cause problem with pointers   
  
(a) Only f3  
(b) Only f1 and f3  
(c) Only f1 and f2  
(d) f1 , f2 ,f3

Answer (c)

4)

abc(char a[]){

a++;

printf("%c",\*a);

a++;

printf("%c",\*a);

}

main(){

char a[100];

a[0]='a';a[1]='b';a[2]='c';a[4]='d';

abc(a);

}

a)bc b)error c)no output d)garbage value

ans: a

5)

void e(int );

main()

{

int a;

a=3;

e(a);

}

void e(int n)

{

if(n>0)

{

e(--n);

printf("%d" , n);

e(--n);

}

}

The output for this program is:   
  
(a) 0 1 2 0  
(b) 0 1 2 1  
(c) 1 2 0 1  
(d) 0 2 1 1

Ans (a)

*6)*

void main(){

   int a=-12;

   a=a>>3;

   printf("%d",a);

}

(a) -4

(b) -3

(c) -2

(d) -96

Answer :( c)

7)

Void f(char\*\*);

main()

{

char \* argv[] = { "ab" ,"cd" , "ef" ,"gh", "ij" ,"kl" };

f( argv );

}

void f( char \*\*p )

{

char\* t;

t= (p+= sizeof(int))[-1];

printf( "%s" , t);

}

The output for this program is:   
  
(a) ab  
(b) cd  
(c) ef  
(d) gh

Ans (b)

8)

*main()*

*{*

*int i=5;*

*printf("%d");*

*}*

*a)error b)0 c)5 d)garbage value*

*ans: c*

9)

|  |
| --- |
| #include<stdio.h>  int f(int \*a, int n)  {    if(n <= 0) return 0;    else if(\*a % 2 == 0) return \*a + f(a+1, n-1);    else return \*a - f(a+1, n-1);  }    int main()  {    int a[] = {12, 7, 13, 4, 11, 6};    printf("%d", f(a, 6));    getchar();    return 0;  } |

(A) -9  
(B) 5  
(C) 15  
(D) 19

Answer (C)

*10)*

#include "string.h"

void main(){

   clrscr();

 printf("%d%d",sizeof("string"),strlen("string"));

getch();

}

(a) 6 6

(b) 7 7

(c) 6 7

(d) 7 6

Answer: (d)

11)

main()

{

Char a[3][3]={{‘a’,’b’,’c’},”pqr”,”xy”};

Printf(“%s\n”,&a[0][0]);

}

1. a b)compilation error c)abcpqrxy

Ans c

*12)*

main( )

{

static int a[ ] = {0,1,2,3,4};

int \*p[ ] = {a,a+1,a+2,a+3,a+4};

int \*\*ptr = p;

ptr++;

printf(“\n %d %d %d”, ptr-p, \*ptr-a, \*\*ptr);

\*ptr++;

printf(“\n %d %d %d”, ptr-p, \*ptr-a, \*\*ptr);

\*++ptr;

printf(“\n %d %d %d”, ptr-p, \*ptr-a, \*\*ptr);

++\*ptr;

printf(“\n %d %d %d”, ptr-p, \*ptr-a, \*\*ptr);

}

***a)*** *111 b) 111 c) 111 d) 111*

*222 222 222 111*

*333 111 333 333*

*344 222 444 444*

*Ans: a*

|  |
| --- |
| 13)  void reverse(void)   {    int c;    if (?1) reverse() ;    ?2  }  main()  {    printf ("Enter Text ") ;    printf ("\n") ;    reverse();    printf ("\n") ;  } |

(A) ?1 is (getchar() != ’\n’)  
?2 is getchar(c);  
(B) ?1 is (c = getchar() ) != ’\n’)  
?2 is getchar(c);  
(C) ?1 is (c != ’\n’)  
?2 is putchar(c);  
(D) ?1 is ((c = getchar()) != ’\n’)  
?2 is putchar(c);

Answer(D)

14)

void main(){

   int a=25;

   clrscr();

   printf("%o %x",a,a);

   getch();

}

(a) 25 25

(b) 025 0x25

(c) 12 42

(d) 31 19

Answer: (d)

15)

#define call(x) #x

void main(){

   printf("%s",call(c/c++));

}

(a)c

(b)c++

(c)#c/c++

(d)c/c++

Answer: (d)

16)

char \*strexp()

         {

         char \*temp = "example string";

         return temp;

         }

         int main()

         {

         puts(strexp);

         }

a)example string b)garbage value c)error

ans:example string

**17) Consider the following C program segment:**

|  |
| --- |
| char p[20];  char \*s = "string";  int length = strlen(s);  int i;  for (i = 0; i < length; i++)       p[i] = s[length — i];  printf("%s",p); |

**The output of the program is (GATE CS 2004)**  
a) gnirts  
b) gnirt  
c) string  
d) no output is printed

Answer(d)

18)

main()

{

Inta[5];

a[-2]=10;

a[2]=1;

printf(“%d”,-2[a]);

}

1. compilation error b)10 c)-1 d)none of the above

Ans c

**Answer:** (c)

**19) The most appropriate matching for the following pairs (GATE CS 2000)**

**X: m=malloc(5); m= NULL; 1: using dangling pointers**

**Y: free(n); n->value=5; 2: using uninitialized pointers**

**Z: char \*p; \*p = ’a’; 3. lost memory is:**

(a) X—1 Y—3 Z-2  
(b) X—2 Y—1 Z-3  
(C) X—3 Y—2 Z-1  
(d) X—3 Y—1 Z-2

Answer (d)

**20) Consider the following C-function in which a[n] and b[m] are two sorted integer arrays and c[n + m] be another integer array.**

|  |
| --- |
| voidxyz(inta[], intb [], intc[])  {    inti, j, k;    i = j = k = O;    while((i<n) && (j<m))       if(a[i] < b[j]) c[k++] = a[i++];       elsec[k++] = b[j++];  } |

**Which of the following condition(s) hold(s) after the termination of the while loop?  
(i) j < m, k = n+j-1, and a[n-1] < b[j] if i = n  
(ii) i < n, k = m+i-1, and b[m-1] <= a[i] if j = m**  
(A) only (i)  
(B) only (ii)  
(C) either (i) or (ii) but not both  
(D) neither (i) nor (ii)

Answer (C)