Q1 What is the output of following program?

|  |
| --- |
| # include <stdio.h>  void fun(int x)  {      x = 30;  }    int main()  {    int y = 20;    fun(y);    printf("%d", y);    return 0;  } |

**Ans:** B. 20

Q2 Output of following program?

|  |  |
| --- | --- |
| # include <stdio.h>  void fun(int \*ptr)  {      \*ptr = 30;  }    int main()  {    int y = 20;    fun(&y);    printf("%d", y);      return 0;  }  **Ans:** B. 30  Q3 Output of following program?   |  | | --- | | #include <stdio.h>    int main()  {      int \*ptr;      int x;        ptr = &x;      \*ptr = 0;        printf(" x = %dn", x);      printf(" \*ptr = %dn", \*ptr);        \*ptr += 5;      printf(" x  = %dn", x);      printf(" \*ptr = %dn", \*ptr);        (\*ptr)++;      printf(" x = %dn", x);      printf(" \*ptr = %dn", \*ptr);        return 0;  } | |

**Ans:** A. x = 0  
 \*ptr = 0  
 x = 5  
 \*ptr = 5  
 x = 6  
 \*ptr = 6

Q4 Consider a compiler where int takes 4 bytes, char takes 1 byte and pointer takes 4 bytes.

|  |  |  |  |
| --- | --- | --- | --- |
| #include <stdio.h>    int main()  {      int arri[] = {1, 2 ,3};      int \*ptri = arri;        char arrc[] = {1, 2 ,3};      char \*ptrc = arrc;        printf("sizeof arri[] = %d ", sizeof(arri));      printf("sizeof ptri = %d ", sizeof(ptri));        printf("sizeof arrc[] = %d ", sizeof(arrc));      printf("sizeof ptrc = %d ", sizeof(ptrc));        return 0;  }  **Ans:** D. sizeof arri[] = 12 sizeof ptri = 4 sizeof arrc[] = 3 sizeof ptrc = 4  Q5 Assume that float takes 4 bytes, predict the output of following program.   |  |  |  | | --- | --- | --- | | #include <stdio.h>    int main()  {      float arr[5] = {12.5, 10.0, 13.5, 90.5, 0.5};      float \*ptr1 = &arr[0];      float \*ptr2 = ptr1 + 3;        printf("%f ", \*ptr2);      printf("%d", ptr2 - ptr1);       return 0;  }  **Ans:** A. 90.500000 3  Doubt   |  | | --- | | **Q6** |  |  | | --- | | #include<stdio.h>  int main()  {      int arr[] = {10, 20, 30, 40, 50, 60};      int \*ptr1 = arr;      int \*ptr2 = arr + 5;      printf("Number of elements between two pointer are: %d.",                                  (ptr2 - ptr1));      printf("Number of bytes between two pointers are: %d",                                (char\*)ptr2 - (char\*) ptr1);      return 0;  } | | |
|  |

**Ans:** A. Number of elements between two pointer are: 5. Number of bytes between two pointers are: 20

|  |
| --- |
| **Q7** |

|  |
| --- |
| #include<stdio.h>  int main()  {     int a;     char \*x;     x = (char \*) &a;     a = 512;     x[0] = 1;     x[1] = 2;     printf("%dn",a);     return 0;  } |

**Ans:** A. Machine dependent

a=512

char\*x means creating a variable x storing the value of char type

Now X = (char\*) &a

This means typecasting int value of a to char value

Since int stores 16 bits but char stores 8 bits

Binary representation of 512 is 00000010 00000000

X will point to 8 bits representation only ( i e 00000000 )

Now we are saying that X[0]=1 this means this 0000000 will change to 00000001 ( this is the binary representation of 1)

X[1]=2 so 2nd set of 8 bit will be Changed to 2 but it was already 2

So finally I get the expression as

00000010 00000001

This is 513

Output - 513n

( n since after %d we have n)

**Q8**

|  |
| --- |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| int main()  {   char \*ptr = "GeeksQuiz";   printf("%cn", \*&\*&\*ptr);   return 0;  }  **Ans:** D. G  **Q9**  #include<stdio.h>  void fun(int arr[])  {    int i;    int arr\_size = sizeof(arr)/sizeof(arr[0]);    for (i = 0; i < arr\_size; i++)        printf("%d ", arr[i]);  }    int main()  {    int i;    int arr[4] = {10, 20 ,30, 40};    fun(arr);    return 0;  }  **Ans:** B. Machine Dependent  Q10 The reason for using pointers in a Cprogram is  **Ans:** D. All of the above  **Q11**   |  | | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | #include<stdio.h>  void f(int \*p, int \*q)  {    p = q;    \*p = 2;  }  int i = 0, j = 1;  int main()  {    f(&i, &j);    printf("%d %d n", i, j);    getchar();    return 0;  }  **Ans:** D. 0 2  Q12 Consider this C code to swap two integers and these five statements after it:   |  | | --- | | void swap(int \*px, int \*py)  {     \*px = \*px - \*py;     \*py = \*px + \*py;     \*px = \*py - \*px;  } |   S1: will generate a compilation error S2: may generate a segmentation fault at runtime depending on the arguments passed S3: correctly implements the swap procedure for all input pointers referring to integers stored in memory locations accessible to the process S4: implements the swap procedure correctly for some but not all valid input pointers S5: may add or subtract integers and pointers.  **Ans:** C. S2 and S4  **Q13**   |  | | --- | |  |  |  | | --- | | int f(int x, int \*py, int \*\*ppz)  {    int y, z;    \*\*ppz += 1;     z  = \*\*ppz;    \*py += 2;     y = \*py;     x += 3;     return x + y + z;  }    void main()  {     int c, \*b, \*\*a;     c = 4;     b = &c;     a = &b;     printf("%d ", f(c, b, a));     return 0;  } |   **Ans:** B. 19  Q14 Predict the output of following program   |  | | --- | | #include<stdio.h>  int main()  {      int a = 12;      void \*ptr = (int \*)&a;      printf("%d", \*ptr);      getchar();      return 0;  }  **Ans:** B. Compiler Error  **Q15** | | #include<stdio.h>    void swap (char \*x, char \*y)  {      char \*t = x;      x = y;      y = t;  }    int main()  {      char \*x = "geeksquiz";      char \*y = "geeksforgeeks";      char \*t;      swap(x, y);      printf("(%s, %s)", x, y);      t = x;      x = y;      y = t;      printf("n(%s, %s)", x, y);      return 0;  }  **Ans:** A. (geeksquiz, geeksforgeeks)  (geeksforgeeks, geeksquiz)  **Q16** | |  | | #include <stdio.h>  int main()  {      int arr[] = {1, 2, 3, 4, 5};      int \*p = arr;      ++\*p;      p += 2;      printf("%d", \*p);      return 0;  }  **Ans:** B. 3  Q17 What does the following program print?   |  | | --- | | #include  void f(int \*p, int \*q)  {    p = q;   \*p = 2;  }  int i = 0, j = 1;  int main()  {    f(&i, &j);    printf("%d %d n", i, j);    getchar();    return 0;  } |   **Ans:** D. 0,2   |  | | --- | | **Q18** |  |  | | --- | | #include <stdio.h>  void f(char\*\*);  int main()  {      char \*argv[] = { "ab", "cd", "ef", "gh", "ij", "kl" };      f(argv);      return 0;  }  void f(char \*\*p)  {      char \*t;      t = (p += sizeof(int))[-1];      printf("%sn", t);  } |   **Ans:** D. gh  Q19 What does the following C-statement declare? [1 mark]   |  | | --- | | int ( \* f) (int \* ) ; |   **Ans:** C. A pointer to a function that takes an integer pointer as argument and returns an integer.   |  |  | | --- | --- | | **………………………………………..DOUBT………………….………………..**   |  | | --- | | **Q20** |   Consider the C program shown below.  #include <stdio.h>  #define print(x) printf("%d ", x)  int x;  void Q(int z)  {      z += x;      print(z);  }  void P(int \*y)  {      int x = \*y + 2;      Q(x);      \*y = x - 1;      print(x);  }  main(void)  {      x = 5;      P(&x);      print(x);  } |   The output of this program is   |  | | --- | |  |   Q21 Suppose that in a C program snippet, followings statements are used.   |  | | --- | | i) sizeof(int);  ii) sizeof(int\*);  iii) sizeof(int\*\*); |   Assuming size of pointer is 4 bytes and size of int is also 4 bytes, pick the most correct answer from the given options. |   **Ans:** B. i), ii) and iii) would compile successfully and size of each would be same i.e. 4   |  | | --- | |  |   Q22 Assume *int* is 4 bytes, *char* is 1 byte and *float* is 4 bytes. Also, assume that pointer size is 4 bytes (i.e. typical case)   |  | | --- | | char \*pChar;  int \*pInt;  float \*pFloat;    sizeof(pChar);  sizeof(pInt);  sizeof(pFloat); |   What’s the size returned for each of sizeof() operator?  **Ans:** A.4 4 4  Q23 In the below statement, ptr1 and ptr2 are uninitialized pointers to int i.e. they are pointing to some random address that may or may not be valid address.   |  | | --- | | int\* ptr1, ptr2; | |   **Ans:** B. FALSE  Q24 Pick the best statement for the following program snippet:   |  | | --- | | #include <stdio.h>    int main()  {   int var;  /\*Suppose address of var is 2000 \*/     void \*ptr = &var;   \*ptr = 5;   printf("var=%d and \*ptr=%d",var,\*ptr);     return 0;  } |   **Ans:** D. Compile error   |  | | --- | |  |   Q25 Consider the following C program.   |  | | --- | | #include<stdio.h>  void mystery(int \*ptra, int \*ptrb)  {     int \*temp;     temp = ptrb;     ptrb = ptra;     ptra = temp;  }  int main()  {      int a=2016, b=0, c=4, d=42;      mystery(&a, &b);      if (a < c)         mystery(&c, &a);      mystery(&a, &d);      printf("%dn", a);  } |   **Ans:** A. 2016   |  | | --- | |  |   Q26 The value printed by the following program is   |  | | --- | | void f(int\* p, int m)  {      m = m + 5;      \*p = \*p + m;      return;  }  void main()  {      int i=5, j=10;      f(&i, j);      printf("%d", i+j);  } |   **Ans:** C. 30  Q27 Consider the C program below. What does it print?   |  | | --- | | # include <stdio.h>  # define swapl (a, b) tmp = a; a = b; b = tmp  void swap2 ( int a, int b)  {          int tmp;          tmp = a; a = b; b = tmp;   }  void swap3 (int\*a, int\*b)  {          int tmp;          tmp = \*a; \*a = \*b; \*b = tmp;  }  int main ()  {          int num1 = 5, num2 = 4, tmp;          if (num1 < num2) {swap1 (num1, num2);}          if (num1 < num2) {swap2 (num1 + 1, num2);}          if (num1 >= num2) {swap3 (&num1, &num2);}          printf ("%d, %d", num1, num2);  } |   **Ans:** C. 4, 5   |  | | --- | |  |   Q28 What will be the output produced by the following C code:  int main()  {  int array[5][5];  printf("%d",( (array == \*array) && (\*array == array[0]) ));  return 0;  }  **Ans:** A. 1   |  | | --- | | **Q29** |   Consider the following C code  int main()  {  int a = 300;  char \*b = (char \*)&a;  \*++b = 2;  printf("%d ",a);  return 0;  }  Consider the size of int as two bytes and size of char as one byte. Predict the output of the following code . Assume that the machine is little-endian.  **Ans:** A. 556   |  | | --- | |  |   Q30 Consider the following function implemented in C:  void printxy(int x, int y)  {  int \*ptr;  x = 0;  ptr = &x;  y = \*ptr;  \*ptr = 1;  printf("%d,%d", x, y);  }  The output of the printxy(1,1) is  **Ans:** C. 1,0  Q31 Consider the following snippet of a C program. Assume that swap(&x, &y) exchanges the contents of x and y.  int main()  {  int array[] = {3, 5, 1, 4, 6, 2};  int done = 0;  int i;  while (done == 0)  {  done = 1;  for (i = 0; i <= 4; i++)  {  if (array[i] < array[i+1])  {  swap(&array[i], &array[i+1]);  done = 0;  }  }  for (i = 5; i >= 1; i--)  {  if (array[i] > array[i-1])  {  swap(&array[i], &array[i-1]);  done = 0;  }  }  }  printf("%d", array[3]);  }  The output of the program is \_\_\_\_\_. Note: This question appeared as Numerical Answer Type.  **Ans:** C. 3   |  | | --- | |  |   Q32. Faster access to non-local variables is achieved using an array of pointers to activation records, called a  **Ans:** D. activation tree  Q33. ‘ptrdata’ is a pointer to a data type. The expression \*ptrdata++ is evaluated as (in C++) :  **Ans:** D. \*(ptrdata++)   |  | | --- | |  |   Q34 Consider the following table   |  |  | | --- | --- | | A. Activation record | p. Linking loader | | B. Location counter | q. Garbage collection | | C. Reference counts | r. Subroutine call | | D. Address relocation | s. Assembler |   **Ans:** C. r, s, q, p  Q35 What does the following C-statement declare? int (\*f) (int\*);  **Ans:** C. A pointer to a function that takes an integer pointer as argument and returns an integer   |  | | --- | |  |   Q36 What is the output of this C code?  #include<stdio.h>  void main()  {  int k=5;  int \*p=&k;  int \*\*m=&p;  printf("%d %d %d",k,\*p,\*\*m);  }  **Ans:** A. 5 5 5   |  | | --- | |  |   Q37 Consider the following declaration:  int a, \*b=&a, \*\*c=&b;  The following program fragment  a=4;  \*\*c=5;  **Ans:** D. assigns 5 to a  Q38 The following ‘C’ statement : int \* f [ ] ( ); declares:  **Ans:** B. Array of functions returning pointers to integers  Q39 Which of the following is true with respect to Reference?  **Ans:** A. A reference can never be NULL  Q40 The following statement in ‘C’  int (\*f())[ ];  declares  **Ans:** A.a function returning a pointer to an array of integers.  DOUBT  Q41 What does the following expression means ?  char ∗(∗(∗ a[N]) ( )) ( );  Q42 What will be the output of following C program?  main()  {  char g[] = "geeksforgeeks";  printf("%s", g + g[6] - g[8]);  }  **Ans:** A. geeks  Q43 What is printed by the following ANSI C program?  #include<stdio.h>  int main(int argc, char \*argv[])  {  int x = 1, z[2] = {10, 11};  int \*p = NULL;  p = &x;  \*p = 10;  p = &z[1];  \*(&z[0] + 1) += 3;  printf("%d, %d, %dn", x, z[0], z[1]);  return 0;  }  **Ans:** D. 10, 10, 14 |