1. What is right way to initialize array?

A.int num[6] = { 2, 4, 12, 5, 45, 5 };

B.int n{} = { 2, 4, 12, 5, 45, 5 };

C.int n{6} = { 2, 4, 12 };

D.int n(6) = { 2, 4, 12, 5, 45, 5 };

1. What will be the output of the program ?

#include<stdio.h>

void main()

{

int a[5] = {5, 1, 15, 20, 25};

int i, j, m;

i = ++a[1];

j = a[1]++;

m = a[i++];

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

}

A.3, 2, 15

B.2, 3, 20

C.2, 1, 15

D.1, 2, 5

1. An array elements are always stored in \_\_\_\_\_\_\_\_ memory locations.

A.Sequential

B.Random

C.Sequential and Random

D.None of the above

1. What is the maximum number of dimensions an array in C may have?

A.2

B.8

C.50

D.Theoratically no limit. The only practical limits are memory size and compilers.

1. What will be printed after execution of the following code?

void main()

{

int arr[10] = {1,2,3,4,5};

printf("%d", arr[5]);

}

A.Garbage Value

B.5

C.6

D.0

1. What will be the output of the following program?

void main()

{

int str1[] = {1,2,3,4};

int str2[] = {1,2,3,4};

if(str1==str2)

printf("Equal");

else

printf("Unequal");

}

A.Equal

B.Unequal

C.Error

D. None of these

1. What will be the output of the following code?

void main()

{

int a[10];

printf("%d %d", a[-1], a[12]);

}

A.0 0

B.Garbage value 0

C.0 Garbage Value

D.Garbage vlaue Garbage Value

1. What will be the output of the program ?

#include

int main()

{

int arr[1] = {10};

printf("%d", 0[arr]);

return 0;

}

A.1

B.0

C.10

D.6

1. What will be the output of the program ?

#include

int main()

{

int arr[1] = {10};

printf("%d", arr[0]);

return 0;

}

A.1

B.0

C.10

D.6

1. What will be the output of the program if the array begins at address 65486 and int takes 2 bytes in memory?

#include<stdio.h>

void main()

{

int arr[] = {12, 14, 15, 23, 45};

printf("%u", arr);

}

A.65486

B. 65488

C. 65492

D. 65482

1. What will be the output of the program if the array begins at address 65486 and int takes 2 bytes in memory?

#include<stdio.h>

void main()

{

int arr[] = {12, 14, 15, 23, 45};

printf("%u", &arr);

}

A.65486

B. 65488

C. 65492

D. 65482

1. What will be the output of the program?

#include

void main()

{

float arr[] = {12.4, 2.3, 4.5, 6.7};

printf("%d", sizeof(arr)/sizeof(arr[0]));

}

A.5

B.4

C.6

D.7

1. Which of the following statements are correct about the program below?

#include<stdio.h>

void main()

{

int size, i;

scanf("%d", &size);

int arr[size];

for(i=1; i<=size; i++)

{

scanf("%d", arr[i]);

printf("%d", arr[i]);

}

}

A.The code is erroneous since the statement declaring array is invalid.

B.The code is erroneous since the subscript for array used in for loop is in the range 1 to size.

C.The code is correct and runs successfully.

D.The code is erroneous since the values of array are getting scanned through the loop.

1. Which of the following statements are correct about an array?

1. The array int num[26]; can store 26 elements.

2. The expression num[1] designates the very first element in the array.

3. It is necessary to initialize the array at the time of declaration.

4. The declaration num[SIZE] is allowed if SIZE is a macro.

A. 1 and 2

B. 1 and 3

C. 1 and 4

D. 2 and 4

1. #include <stdio.h>

int main(void) {

int i;

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

for(i=5;i>=2;i--)

a[i-1]=a[i-2];

for(i=0;i<5;i++)

printf("%d ",a[i]);

return 0;

}

1 1 2 3 4

A.1

B.1, 4

C.2, 3

D.2, 4

1. What is meaning of following declaration ?

int arr[20];

A Array of size 20 that can have integer address

B. Integer Array of size 20

C None of these

D Array of Size 20

1. int a[20]

What will be the size of above array element ?

A 21

B 19

C 20

D 22

1. Below is an example of -

int RollNum[30][4];

A 3-D Array

B 4-D Array

C 1-D Array

D 2-D Array

1. How many elements are there in array a[3][4][5]?

A. 24

B. 60

C. 12

D. 20

1. Which of the following correctly declares an array?

A. int anarray[10];

B. int anarray;

C. anarray{10};

D. array anarray[10];

1. What is the index number of the last element of an array with 29 elements?

A. 29

B. 28

C. 0

D. Programmer-defined

1. Which of the following correctly accesses the seventh element stored in foo, an array with 100 elements?

A. foo[6];

B. foo[7];

C. foo(7);

D. foo;

1. What is the output of this C code?

#include <stdio.h>

void main()

{

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

int i = 0, j = 0;

for (i = 0; i < 2; i++)

for (j = 0; j < 3; j++)

printf("%d", a[i][j]);

}

a) 1 2 3 4 5 0

b) 1 2 3 4 5 junk

c) 1 2 3 4 5 5

d) Run time error

1. What is the output of this C code?

#include <stdio.h>

void main()

{

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

int i = 0, j = 0;

for (i = 0; i < 2; i++)

for (j = 0; j < 3; j++)

printf("%d", a[i][j]);

}

a) 1 2 3 junk 4 5

b) Compile time error

c) 1 2 3 0 4 5

d) 1 2 3 3 4 5

1. Consider the following declaration of a ‘two-dimensional array in C:

char a[100][100];

Assuming that the main memory is byte-addressable and that the array is stored starting from memory address 0, the address of a[40][50] is (GATE CS 2002)

(A) 4040

(B) 4050

(C) 5040

(D) 5050

1. Predict output of following program

int main()

{

int i;

int arr[5] = {1};

for (i = 0; i < 5; i++)

printf("%d ", arr[i]);

return 0;

}

A 1 followed by four garbage values

B 1 0 0 0 0

C 1 1 1 1 1

D 0 0 0 0 0

1. What is the output of the following program?

int main()

{

int i;

int arr[5] = {0};

for (i = 0; i <= 5; i++)

printf("%d ", arr[i]);

return 0;

}

A. Compiler Error: Array index out of bound.

B The always prints 0 five times followed by garbage value

C The program always crashes.

D. The program may print 0 five times followed by garbage value, or may crash if address (arr+5) is invalid.

1. Consider the C program given below. What does it print?

#include <stdio.h>

int main ()

{

int i, j;

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

for(i = 0; i < 3; i++)

{

a[i] = a[i] + 1;

i++;

}

printf ("%d, %d", i, a[i]);

}

A. 2,3

B. 4,5

C. 3,4

D. 4 4

1. Consider the C program given below. What does it print?

#include <stdio.h>

int main ()

{

int i, j;

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

for(i = 0; i < 3; i++)

{

a[i] = a[i] + 1;

i+=2;

}

printf ("%d, %d", i, a[i]);

}

A. 2,3

B. 4,5

C. 3,4

D. 4 4

1. A(n) ..........is an integral value used to access an element in an array

Constant

element

index

number

1. Which of following statements assigns the value stored in x to the first element on an array, ary?

a. ary=x;

b. ary=x]0];

c. arr[0]=x;

d. ary=x[1];

1. The process through which data are arranged according to their values is known as

arranging

listing

parsing

sorting

1. #include <stdio.h>

int main ()

{

int i;

int list[10]={0};

for(i=0;i<5;i++)

list[2\*i+1]=list[i+2];

for(i=0;i<10;i++)

printf("%d ",list[i]);

return 0;

}

A.0 2 0 3 0 4 0 5 0 6

B. 0 1 0 2 0 3 0 4 0 5

C. 0 0 0 0 0 0 0 0 0 0

D. 0 2 0 4 0 4 0 5 0 5

1. #include <stdio.h>

int main ()

{

int i;

int list[10]={0};

for(i=0;i<5;i++)

list[2\*i+1]=i+2;

for(i=0;i<10;i++)

printf("%d ",list[i]);

return 0;

}

A.0 2 0 3 0 4 0 5 0 6

B. 0 1 0 2 0 3 0 4 0 5

C. 0 0 0 0 0 0 0 0 0 0

D. 0 2 0 4 0 4 0 5 0 5

1. An array is declared as float arr[]={1,3,5,7,9}; then what is the value of sizeof(arr[3])?

A. 4

B. 7

C. 2

D. Garbage Value

1. If an array is declared as arr[]={1,3,5,7,9}; then what is the value of arr[3]?

A. 4

B. 7

C. 2

D. Garbage Value

1. If an array is declared as double arr[50], how many bytes will be allocated to it?

A. 200

B. 100

C. 300

D. 400

1. If an array is declared as char arr[50], how many bytes will be allocated to it?

A. 200

B. 100

C. 50

D. 400

1. If an array is declared as float arr[50], how many bytes will be allocated to it?

A. 200

B. 100

C. 50

D. 400

1. If an array is declared as float arr[5][2], how many bytes will be allocated to it?

A. 20

B. 10

C. 50

D. 40

1. If an array is declared as char arr[5][2], how many bytes will be allocated to it?

A. 20

B. 10

C. 50

D. 40

1. If the size of the array is less than the number of initializers

then,

A ] Extra values are being ignored

B ] Generates an error message

C ] Size of Array is increased

D ] Size is neglected when values are given

1. Array subscripts in C always start at

A ] -1

B] 1

C] 0

D] Value provided by user

1. #include <stdio.h>

int main(void) {

int i;

int a[5]={2,4,6,8,10};

for(i=4;i>=0;i=i-1)

printf("%d ",a[i]);

return 0;

}

A. 2 4 6 8 10

B. 10 8 6 4 2

C. 2 10 2 8 6

D. None of these

1. #include <stdio.h>

int main(void) {

int i;

int a[5]={2,4,6,8,10};

for(i=4;i>=0;i=i-2)

printf("%d ",a[i]);

return 0;

}

A. 2 6 10

B. 10 6 2

C. 2 2 6

D. None of these

1. int main(void) {

int i;

int a[5]={2,4,6,8,10};

for(i=4;i>=0;i=i>>1)

printf("%d ",a[i]);

return 0;

}

A. 10 8 6 2 0

B. 10 8 6 4

C. Infinite loop

D. None of these

1. #include <stdio.h>

int main(void) {

int i;

int a[5]={2,4,6,8,10};

for(i=4;i>=0;i--)

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

return 0;

}

A. 6 4 4 2 2

B. 10 8 6 4 2

C. 2 4 6 8 10

D. None of these

1. #include <stdio.h>

int main(void) {

int i;

int a[5]={2,4,6,8,10};

for(i=4;i>=0;i--)

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

return 0;

}

A. garbage garbage 10 6 2

B. Garbage 8 4 6 2

C. 10 8 6 4 2

D. 2 4 6 8 10

1. #include <stdio.h>

int main(void) {

int i;

int a[5]={2,4,6,8,10};

for(i=4;i>=0;i--)

printf("%d ",++a[i]);

return 0;

}

A. 11 9 7 5 3

B. Error

C. 10 8 6 4 2

D. None of these

1. #include <stdio.h>

int main(void) {

int i;

int a[5]={2,4,6,8,10};

for(i=1;i<=2;i++)

printf("%d ",a[i]);

return 0;

}

A. 4 6

B. 2 4 6

C. 8 6 4

D. 4 6 8

1. #include <stdio.h>

int main(void) {

int i;

int a[5]={2,4,6,8,10};

for(i=2;i>=2;i--)

printf("%d ",a[i]);

return 0;

}

A. 6

B. 8

C. 4 6

D. 8 6

1. #include <stdio.h>

int main(void) {

int i;

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

for(i=5;i>=2;i--)

a[i-1]=a[i-2];

for(i=0;i<5;i++)

printf("%d ",a[i]);

return 0;

}

A. 1 1 2 3 4

B. 1 2 2 3 4

C. 1 2 3 3 4

D. 1 2 3 4 4

1. #include <stdio.h>

int main(void) {

int i;

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

for(i=5;i>=2;i--)

a[i+1]=a[i-2];

for(i=0;i<5;i++)

printf("%d ",a[i]);

return 0;

}

A. 1 1 2 3 4

B. 1 2 2 3 4

C. 1 2 3 1 2

D. 1 2 3 4 4

1. #include <stdio.h>

int main(void) {

int i;

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

for(i=1;i<=4;i++)

a[i]=a[i-2];

for(i=0;i<5;i++)

printf("%d ",a[i]);

return 0;

}

1 junk 1 junk 1

1 2 3 4 5

2 3 4 5 junk

1 2 3 4 junk

1. #include <stdio.h>

int main(void) {

int i;

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

for(i=1;i<=4;i++)

a[i-2]=a[i];

for(i=0;i<5;i++)

printf("%d ",a[i]);

return 0;

}

3 4 5 4 5

1 2 3 2 3

4 3 3 5 5

1 2 3 4 5

1. #include <stdio.h>

int main(void) {

int i;

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

for(i=0;i<5;i++)

printf("%d ",a[i]++);

return 0;

}

1 2 3 4 5

2 3 4 5 6

2 3 4 5 6 7

None of these

1. #include <stdio.h>

int main(void) {

int i;

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

for(i=0;i<5;i++);

printf("%d ",a[i]++);

return 0;

}

1 2 3 4 5

2 3 4 5 6

2 3 4 5 6 7

None of these

1. #include <stdio.h>

int main(void) {

int i;

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

for(i=0;i<5;i++)

printf("%d ",++a[i]);

return 0;

}

1 2 3 4 5

2 3 4 5 6

2 3 4 5 6 7

None of these

1. #include <stdio.h>

int main(void) {

int i;

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

for(i=4;i>=0;i--)

printf("%d ",++a[i]);

return 0;

}

1 2 3 4 5

5 4 3 2 1

2 3 4 5 6

6 5 4 3 2

1. Given the base address of an array **B[1300…..1900]** as 1020 and size of each element is 2 bytes in the memory. Find the address of **B[1700]**.

A.

B.1820

C.

D.

1. Consider a 3 x 4 integer array A . Assume base address is 1000 and each element takes 2 bytes in memory. Find the address of A[3][2] using column major.

A. 1012

B. 1010

C. 1016

D. 1018

1. Consider a 3 x 4 integer array A . Assume base address is 1000 and each element takes 2 bytes in memory. Find the address of A[3][2] using row major.

A. 1012

B. 1010

C. 1016

D. 1018

1. Suppose element of array A[4][5] occupies 4 bytes, and the address of the 1st element is 49.  Find the address of the element A(4,3) when the storage is row major.

A.151

B. 149

C. 141

D. 98