WORKSHEET – ARRAYS-Sheet-9

1. main()

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

printf("%d %d %d",arr[1],arr[2],arr[3]);

}

2. main()

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

arr[0,1,2]=10;

printf("%d%d%d",arr[0],arr[1],arr[2]);

}

3. main()

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

arr[1+2]=10;

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

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

}

4. main()

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

arr[2.5+1.5]=10;

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

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

}

5. main()

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

printf("The number of elements in array are %d",sizeof(array)/sizeof(array[0]));

}

6. main()

{int a=10,b;

int arr[]={1,2,3},brr[3];

printf("Assigning the content of a to b\n");

b=a;

printf("Assigning the contents of one array to another\n");

brr=arr;

printf("Contents of brr are %d%d%d",brr[0],brr[1],brr[2]);

}

7. main()

{int arr[]={1,2,3},brr[]={1,2,3};

if(arr==brr)

printf("Contents of array arr and brr are same\n");

else

printf("Contents of array arr and brr are not same");

}

8. main()

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

// starting address from 2000

int \*ptr=a;

printf("%d %d\n%u %u ",\*a,\*ptr,a,ptr);

}

9. main()

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

// starting address from 2000

printf("%d%d\n",arr,&arr);

printf("%d%d",++arr,++&arr);

}

10. main()

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

// starting address from 2000

printf("%d%d\n",arr,&arr);

printf("%d%d",arr+1,&arr+1);

}

11. main()

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

printf("%d%d%d%d%d",\*a,\*(a+0),\*(0+a),a[0],0[a]);

}

12. main()

{int \*ptr;

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

// starting address from 2000

ptr=arr;

printf("%d%d",arr[2],ptr[2]);

}

13. main()

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

// starting address from 2000

int j,\*ptr=arr;

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

{printf("%d",\*ptr);

++ptr;}

}

14. main()

{int j=20;

int arr[]={10,30,40,50}; //starting address from 2000

int i,\*ptr;

ptr=arr;

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

{printf("%d",\*ptr);

ptr++;

}}

15. main()

{

int arr[]={1,2,3,4,5}; // Starting address 2000

int \*ptr=arr;

ptr=ptr+1;

printf("The value pointed by ptris%d",\*ptr);

}

16. main()

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

// Starting address 2000

int \*ptr1=arr;

int \*ptr2=arr+3;

printf("The result of ptr2-ptr1 is%d",ptr2-ptr1);

}

17. main()

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

int \*ptr1=arr;

int \*ptr2;

ptr2=ptr1\*2;

printf("The value of ptr2 is%d",ptr2);

}

18.int size = 4;

int arr[size];

int main()

{

if(arr[0])

printf("Initialized to ZERO");

else

printf("Not initialized to ZERO");

}

19.int main()

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

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

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

return 0;

}

1. int main()

{ int i;

int arr[5] = {1};

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

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

return 0;

}

1. int main()

{

int i;

int arr[5] = {0};

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

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

return 0;

}

1. int main()

{

char p;

char buf[10] = {1, 2, 3, 4, 5, 6, 9, 8};

p = (buf + 1)[5];

printf("%dn", p);

return 0;

}

1. int \* arrPtr[5];

int main()

{

if(\*(arrPtr+2) == \*(arrPtr+4))

{

printf("Equal!");

}

else

{

printf("Not Equal");

}

}

1. #include <stdio.h>

int main()

{

int arr[50] = {0,1,2,[47]=47,48,49};

int i;

for(i=0;i<50;i++)

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

return 0;

}

1. #include <stdio.h>

int main () {

int sum = 0, maxsum = 0, i, n = 6;

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

for (i = 0; i< n; i++) {

if (i == 0 || a [i] < 0 || a [i] < a [i - 1]) {

if (sum >maxsum) maxsum = sum;

sum = (a [i] > 0) ? a [i] : 0;

}

else sum += a [i];

}

if (sum >maxsum) maxsum = sum ;

printf ("%d\n", maxsum);

}

1. 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++;

}

i--;

for (j = 7; j > 4; j--) {

int i = j/2;

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

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

}

1. int main()

{

int i=4,j ,arr[4]={2,4,6,16};

while(i)

{

j=arr[i]+i;

i--;

}

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

}

28. void main( ) {

int x []= {10,20,30,40,50};

print f (“ \n %d %d %d %d %d “, x [4] ,3[x]

,x[2] ,1[x] ,x[0] );

}

29. int main()

{

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

int i, j, m;

i = a[2]++;

j = ++a[1];

m = a[++i];

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

return 0;

}

30. int main()

{

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

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

}

31.main ( )

{

int a, b=0;

static int c [10]={1,2,3,4,5,6,7,8,9,0};

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

if ((c[a]%2) == 0)

b+= c [a];

printf (“%d”, b);

}

32. int main()

{

int i=0,j=0 ,arr[6]={2,4,6,0,5,16};

while(arr[i])

{

j=j+arr[i];

i++;

}

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

}

33. int main()

{

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

for(i=0;i<8;i++)

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

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

return 0;

}

34. What is the output of this C code?

#include <stdio.h>

int main()

{

int arr[5]={5,10,15,20,25};

func(arr);

}

func(int arr[])

{

int i=5,sum=0;

while(i>2)

sum=sum+arr[--i];

printf("%d",sum);

return 0;

}

35. What is the output of this C code?

#include<stdio.h>

swap(int,int);

swap1(int\*,int\*);

int main()

{

int a=4,b=6, arr[5]={1,2,3,4,5};

int a2[5]={6,7,8,9,10};

swap(a,b);

swap1(arr,a2);

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

printf("%d %d\n",arr[0],arr[4]);

printf("%d %d\n",a2[0],a2[4]);

}

swap(int a, int b)

{

int temp;

temp=a,a=b,b=temp;

}

swap1(int arr1[5], int arr2[5] )

{

int i,temp;

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

{

temp=arr1[i];

arr1[i]=arr2[i];

arr2[i]=temp;

}

}

36. What is the output of this C code?

int main()

{

int a[10]={2,-3,4,-5,6,7,1,9,-10,-11};

int i, j,x,k=0;

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

{

x=a[k];

if(x<0)

{

for(j=k;j<10;j++)

a[j]=a[j+1];

a[9]=x;

}

else

k++;

}

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

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

}

37. Tthe array begins 1200 in memory

int main()

{

int arr[]={2, 3, 4, 1, 6};

printf("%u, %u, %u\n", arr, &arr[0], &arr);

return 0;

}

38. main ( )

{

int a, b=0;

static int c [10]={1,2,3,4,5,6,7,8,9,0};

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

if ((c[a]/2)== 4)

b=c[a];

printf ("%d", b);

}

39. int main()

{

int arr[5], i=0;

while(i<5)

arr[i]=++i;

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

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

return 0; }

40. #include<stdio.h>

int main()

{

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

int i;

for (i = 0; i<=4; i++)

if ((char)a[i] == '4')

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

else

printf("FAIL\n");

}

41. main()

{

int arr [];

arr[0]=arr[1]=arr[2]=5;

printf("%d%d%d",arr[0],arr[1],arr[2]);

}