Sheet-10

1. int main()

{

int i=0,j=0;

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

while(arr[i])

{

j=j+arr[i];

i++;

}

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

return 0;

}

1. int main()

{

intary[2][3];

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

printf("%d\n", ary[1][0]);

}

1. void print(int arr[])

{

int n = sizeof(arr)/sizeof(arr[0]);

int i;

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

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

}

int main()

{

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

print(arr);

return 0;

}

1. #define SIZE(arr) sizeof(arr) / sizeof(\*arr);

void fun(int\* arr, int n)

{

int i;

\*arr += \*(arr + n - 1) += 10;

}

void printArr(int\* arr, int n)

{

int i;

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

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

}

int main()

{

int arr[] = {10, 20, 30};

int size = SIZE(arr);

fun(arr, size);

printArr(arr, size);

return 0;

}

1. int main()

{

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

int i, j;

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

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

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

return 0;

}

1. int main()

{

int a[10][20][30] = {0};

a[5][2][1] = 2;

return 0;

}

1. main()

{

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

int i,j,k;

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

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

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

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

}

1. main()

{

int arr[][3]={1,2,3,4}; printf("%d%d%d%d%d%d",arr[0][0],arr[0][1],arr[0][2],arr[1][0],arr[1][1],arr[1][2]);

}

1. main()

{

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

printf("%p %p %p %p " , &arr[0][0],&arr[0][1],&arr[1][0],&arr[1][1]);

}

1. main()

{

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

printf("%d%d%d",arr[1][2],1[arr][2],\*(\*(arr+1)+2));

}

1. main()

{

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

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

}

1. main()

{

int a[]={0,1,2,3,4};// Starting address 1000

int \*p[]={a,a+1,a+2,a+3,a+4}; // Starting address 2000

int \*\*ptr=p;// Starting address 3000

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

}

1. #include <stdio.h>

void main()

{

int a[2][2][2]={1,2,3,4,5,6,7,8};// Starting address 2000

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

}

1. main()

{

void a,b;

void \*ptr;

ptr=&a;

printf("ptr points to a\n");

ptr=&b;

printf("ptr now points to b"); }

1. main()

{

int a=10;

int \*i\_ptr=&a;

void \*v\_ptr=i\_ptr;

\*i\_ptr++;

\*v\_ptr++;

printf("The value of objects pointed to by pointers are%d%d",\*i\_ptr,\*v\_ptr);

}

1. int main()

{

static char s[25] = "The

cocaine man";

int i=0;

char ch;

ch = s[++i];

printf("%c", ch);

ch = s[i++];

printf("%c", ch);

ch = i++[s];

printf("%c", ch);

ch = ++i[s];

printf("%c", ch);

return 0;

}

1. int main()

{

int ary[2][3];

foo(ary); }

void foo(int \*ary[])

{

inti = 10, j = 2, k;

ary[0] = &i;

ary[1] = &j;

\*ary[0] = 2;

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

printf("%d\n", \*ary[k]);

}

1. 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]);

}

1. int main()

{

int arr[10],a[3][4],z[2][3][5];

printf("%u\n%u\n%u\n",sizeof(arr),sizeof(a),sizeof(z));

}

1. int main()

{

int i,j,arr[3][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12}};

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

{

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

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

printf("\n");

}

printf("%d",\*(\*(arr+1)+2));

}

1. void f(int a[][])

{

a[0][1] = 3;

int i = 0, j = 0;

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

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

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

}

void main()

{

int a[2][3] = {0};

f(a);

}

1. void f(int a[][3])

{

a[0][1] = 3;

int i = 0, j = 0;

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

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

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

void main()

{

int a[2][3] = {0};

f(a);

}

1. 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:

(A) 4040

(B) 4050

(C) 5040

(C) 5050

1. int main()

{

foo(ary);

}

void foo(int \*\*ary)

{

int i = 10, k = 10, j = 2;

int \*ary[2];

ary[0] = &i;

ary[1] = &j;

printf("%d\n", ary[0][1]);

}

1. int main()

{

int ary[2][3][4], j = 20;

ary[0][0] = &j;

printf("%d\n", \*ary[0][0]);

}

1. int main()

{

int ary[2][3];

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

printf("%d\n", ary[1][0]);

}

1. int main()

{

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

int \*p = ary + 3;

printf("%d\n", p[-2]);

}

1. int main()

{

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

int \*p = ary + 3;

printf("%d %d\n", p[-2],

ary[\*p]);

}

1. int main() // assume starting address 2000

{

unsigned int x[4][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}};

printf("%u, %u, %u", x+3,

\*(x+3), \*(x+2)+3);

}

1. int main()  
   {  
       int a[3][3][3]= {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27};  
       printf("%u\n",a[1]);  
       printf("%u\n",a[2]);  
       printf("%u\n",a[3]);  
      }
2. int main()  
   {   
       int a[3][3][3]=

{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23, 24,25,26,27}; // starting address 2000  
    printf("%u\n",\*\*a+1);  
    printf("%u\n",\*a+1);  
    printf("%u\n",a+1);  
    printf("%u\n",&a+1);  
  }

1. int main()  
   {  
       int a[3][3][3]= {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27};   
       printf("%u\n",\*\*(a+1));  
       printf("%u\n",\*(a+1));

}

1. int main()  
   {  
       int a[3][3][3]= {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27};   
       printf("%u\n",\*(\*a+1)+1);  
       printf("%u\n",(\*(\*a+1))+1);  
       printf("%u\n",(\*\*a+2)+1);

}

1. int main()  
   {  
       int a[3][3][3]= {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27};    
       printf("%u\n",\*\*(\*a+1)+1);  
       printf("%u\n",\*(\*(\*a+1))+1);  
       printf("%u\n",\*(\*\*a+2)+1);  
   }
2. void main()

{

int a[3][3][3]= {1,[2][2][2]=100};  
    for (int i=0;i<=25;i++)  
    \*(\*\*a+i)=i;  
    for (int i=0;i<3;i++)  
    for (int j=0;j<3;j++)  
    for (int k=0;k<3;k++)  
        printf("%d ",a[i][j][k]);

}