|  |
| --- |
| #include <stdio.h> #include <stdlib.h>    void print\_2d(int arr[][3]){  int i, j;  for (i = 0; i < 3; i++)  for (j = 0; j < 3; j++)  printf("%d ", arr[i][j]); }  void swap\_v1(int a , int b){  int temp;  temp = a;  a = b;  b = temp; }  void swap\_v2(int \* a , int \* b){  int temp;  temp = \*a;  \*a = \*b;  \*b = temp; }   */\* 5.4 start\*/* #define ALLOCSIZE 10000 */\* size of available space \*/* static char array[ALLOCSIZE]; */\* storage for alloc \*/* static char \*allocp = array; */\* next free position \*/*   char \*alloc(int n){ */\* return pointer to n characters \*/*  if (array + ALLOCSIZE - allocp >= n) { */\* it fits \*/*  allocp += n;  return allocp - n; */\* old p \*/*  }  else */\* not enough room \*/*  return 0; } void afree(char \*p) {*/\* free storage pointed to by p \*/*  if (p >= array && p < array + ALLOCSIZE)  allocp = p; } */\* 5.4 end\*/*   int main(){     */\* 5.8 start \*/*   *// 5.8.1 , how to initialize 2d array*  */\*  // method 1   int a[2][3] = {{10, 20, 30}, {25, 26, 27}};  printf("a[0][2] = %d\n" , a[0][2]);  // method 2   int b[][3] = {{10, 20, 30}, {25, 26, 27}};  printf("b[0][2] = %d\n" , b[0][2]);  // method 3   int c[2][3] = {10, 20, 30, 25, 26, 27};  printf("c[0][2] = %d\n" , c[0][2]);   // 5.8.2 , 2d array and functions   int arr[][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  print\_2d(arr);  \*/*  */\* 5.8 end \*/*         */\* 5.7 start\*/*  */\*  int x = 5;  int \* p = &x;  \*p = 6;  int \*\* q = &p;  int \*\*\* r = &q;   printf("%d\n",\*p);  printf("%d\n",\*q);  printf("%d\n",\*\*q);  printf("%d\n",\*\*r);  printf("%d\n",\*\*\*r);  \*\*\*r = 10;  printf("x = %d\n",x);  \*/*  */\* 5.7 end \*/*       */\* 5.6 start \*/*  */\*  int n = 1026; // 0000 0100 0000 0010  printf("%d\n",(char)n);   // with pointer   int \* pN;  pN = &n; // pointer to n  char \* cPN;  cPN = (char \*)pN; // character pointer to n  printf("%d\n" , \*cPN);   void \* vp;  vp = pN;  //printf("%d\n",\*vp); // compile error  printf("%d\n",\*(int \*)vp);  \*/*  */\* 5.6 end /\*     /\* 5.5 \*/*  */\*  char amessage[] = "now is the time"; //an array  char \*pmessage = "now is the time"; // a pointer  printf("%s\n" , amessage);  printf("%s\n" , pmessage);  //amessage = "Hello!"; // wrong!  pmessage = "Hello!";  printf("%s\n" , pmessage);  \*/*  */\* 5.5 end \*/*   */\* 5.4 start \*/*   *// alloc and afree 5.4.1*  *//char \* first\_allocation = alloc(15);*  *//char \* second\_allocation = alloc(7);*  *//afree(first\_allocation);*  *//afree(second\_allocation);*    *//malloc and free*  *//int \*ptr;*  *//ptr = (int \*)malloc(sizeof(int) \* 10);*  *//int i;*  *//for(i=0 ; i<10 ; i++)*  *//\*(ptr + i) = i;*  *//for(i=0 ; i<10 ; i++)*  *//printf("%d " , \*(ptr + i));*    *//char \* chPtr;*  *//chPtr = (char \*)malloc(sizeof(char) \* 10);*  *//int j;*  *//for(j=0 ; j<10 ; j++)*  *//\*(chPtr + j) = j+65;*  *//for(j=0 ; j<10 ; j++)*  *//printf("%c " , \*(chPtr + j));*  */\* 5.4 end \*/*    */\* 5.3 \*/*  */\*  int i;  int ages[5] = {5,12,33,60,64};  for(i = 0 ; i<5 ; i++){  printf("ages[%d] \t %d \t %x\n" , i , ages[i] , &ages[i]);  }   printf("address of ages : %x\n" , ages);  printf("value of age[0] : %d\n" , \*ages);  printf("value of age[1] : %d\n" , \*(ages + 1));  printf("value of age[2] : %d\n" , \*(ages + 2));  printf("value of age[3] : %d\n" , \*(ages + 3));  \*/*     */\* 5.2  int a = 12;  int b = 35;  printf("a=%d b=%d\n" , a , b);  swap\_v1(a,b); // wrong!  printf("a=%d b=%d\n" , a , b);  swap\_v2(&a,&b); // correct!  printf("a=%d b=%d\n" , a , b);  return 0;  \*/*      */\* 5.1 start \*/*  */\*  int temp = 25;  printf("Name\tValue\tAddress\n");  printf("temp\t%d\t%X\n",temp,&temp);  int \* pTemp = &temp;  printf("temp\t%d\t%x\n",temp,pTemp);  printf("pTemp\t%x\t%x\n",pTemp,&pTemp);  \*/*  */\* 5.1 end \*/*   return 0; } |