|  |
| --- |
| Homework : \_\_6A\_\_\_\_  //Christopher Badolato CH432391  //Assignment 6A  //11/8/2018  //ENC3211  //Create a union print out the value as, char, short, int, long.  //for each type we print the value of each type as well as each type in hex.  #include<stdio.h>  union integer{  char c;  short s;  int i;  long b;  };  int main()  {  //initialize our union  union integer unionValue;  //grab or first char and store it on our union.  //it will store the same char in each position as different types.  printf("Enter data type char: ");  scanf("%c", &unionValue.c);  //Print the types of the entered char.  printf("\nBreak down of elements in the union\n");  printf("char %c\n", unionValue.c);  printf("short:%hd\n", unionValue.s);  printf("int %d\n", unionValue.i);  printf("long %ld\n", unionValue.b);  //print the hex of each type.  printf("\nBreak down in hex\n");  printf("char %x\n", unionValue.c);  printf("short:%x\n", unionValue.s);  printf("int %x\n", unionValue.i);  printf("long %x\n", unionValue.b);  //do it for a short  printf("\nEnter data type short: ");  scanf("%hd", &unionValue.s);  printf("\nBreak down of elements in the union\n");  printf("%hd shown as a char: %c\n", unionValue.c);  printf("%hd shown as a short: %hd\n", unionValue.s);  printf("%hd shown as an int: %d\n", unionValue.i);  printf("%hd shown as a long: %ld\n", unionValue.b);  //hex  printf("\nBreak down in hex\n");  printf("char %x\n", unionValue.c);  printf("short:%x\n", unionValue.s);  printf("int %x\n", unionValue.i);  printf("long %x\n", unionValue.b);  //do it for an integer  printf("\nEnter data type integer: ");  scanf("%d", &unionValue.i);  printf("\nBreak down of elements in the union\n");  printf("char %c\n", unionValue.c);  printf("short %hd\n", unionValue.s);  printf("int %d\n", unionValue.i);  printf("long:%ld\n", unionValue.b);  //hex  printf("\nBreak down in hex\n");  printf("char %x\n", unionValue.c);  printf("short:%x\n", unionValue.s);  printf("int %x\n", unionValue.i);  printf("long %x\n", unionValue.b);  //do it for a long  printf("\nEnter data type Long ");  scanf("%ld", &unionValue.b);  printf("\nBreak down of elements in the union\n");  printf("char: %c\n", unionValue.c);  printf("short: %hd\n", unionValue.s);  printf("int: %d\n", unionValue.i);  printf("long: %ld\n", unionValue.b);  //hex  printf("\nBreak down in hex\n");  printf("char %x\n", unionValue.c);  printf("short:%x\n", unionValue.s);  printf("int %x\n", unionValue.i);  printf("long %x\n", unionValue.b);  return 0;  } |
|  |
| Enter data type char: 8  Break down of elements in the union  char 8  short:56  int 56  long 56  Break down in hex  char 38  short:38  int 38  long 38  Enter data type short: 8  Break down of elements in the union  8 shown as a char: ö  8 shown as a short: -108  8 shown as an int: 6356884  8 shown as a long: 6356884  Break down in hex  char 8  short:8  int 8  long 8  Enter data type integer: 8  Break down of elements in the union  char  short 8  int 8  long:8  Break down in hex  char 8  short:8  int 8  long 8  Enter data type Long 8  Break down of elements in the union  char:  short: 8  int: 8  long: 8  Break down in hex  char 8  short:8  int 8  long 8  Process returned 0 (0x0) execution time : 2.927 s  Press any key to continue. |
| Homework 6B  I forgot to change the Assignment name to 6B but it is the correct code! 😊  //Christopher Badolato CH432391  //Assignment 6A  //11/8/2018  //ENC3211  //Create a union print out the value as, float, long and long double  //for each type we print the value of each type as well as each type in hex.  #include <stdio.h>  union floatingPoint{  float f;  double d;  long double l;  };  int main(void)  {  //creates union  union floatingPoint unionValue;  //grabs value from user  printf("Enter data for type float: ");  scanf("%f", &unionValue.f);  //prints the values of the union in each type.  printf("\nBreak down of elements in the union\n");  printf("float %f\n", unionValue.f);  printf("double %lf\n", unionValue.d);  printf("long double %lf\n", unionValue.l);  //in hex  printf("\nBreak down in hex\n");  printf("float %x\n", unionValue.f);  printf("double %x\n", unionValue.d);  printf("long double %x\n", unionValue.l);  //double  printf("\nEnter data for type double: ");  scanf("%lf", &unionValue.d);  printf("\nBreak down of elements in the union\n");  printf("float %f\n", unionValue.f);  printf("double %lf\n", unionValue.d);  printf("long double %Lf\n", unionValue.l);  //in hex  printf("\nBreak down in hex\n");  printf("float %x\n", unionValue.f);  printf("double %x\n", unionValue.d);  printf("long double %x\n", unionValue.l);  //Long double  printf("\nEnter data for type long double: ");  scanf("%Lf", &unionValue.l);  printf("\nBreak down of elements in the union\n");  printf("float %f\n", unionValue.f);  printf("double %lf\n", unionValue.d);  printf("long double %Lf\n", unionValue.l);  //in hex  printf("\nBreak down in hex\n");  printf("float %x\n", unionValue.f);  printf("double %x\n", unionValue.d);  printf("long double %x\n", unionValue.l);  return 0;  } |
|  |
| Enter data for type float: 234.567  Break down of elements in the union  float 234.567001  double 0.000000  long double 0.000000  Break down in hex  float e0000000  double 436a9127  long double 436a9127  Enter data for type double: 234.567  Break down of elements in the union  float -788598326743269380.000000  double 234.567000  long double 234.567000  Break down in hex  float 0  double dd2f1aa0  long double dd2f1aa0  Enter data for type long double: 234.567  Break down of elements in the union  float -788598326743269380.000000  double 234.567000  long double 234.567000  Break down in hex  float 0  double dd2f1aa0  long double dd2f1aa0  Process returned 0 (0x0) execution time : 8.990 s  Press any key to continue. |
| Homework 6C  //Christopher Badolato CH432391  //Assignment 6A  //11/8/2018  //ENC3211  /\*  This program will take in four user entered characters and create one 32-bit  value with them as an unsigned type. We will shift on the letters to the binary  value in the order they were received.  \*/  #include <stdio.h>  //function prototypes  unsigned packCharacters(unsigned ch1, char ch2);  void display(unsigned result);  int main(){  char k, l, m, n;  unsigned result = 1 << 32, result1, result2, result3;  int c, shiftValue = 8;  //get the characters from the user.  printf("Enter first character: ");  scanf( "%c", &k);  printf("Enter second character: ");  scanf(" %c", &l);  printf("Enter third character: ");  scanf(" %c", &m);  printf("Enter fourth character: ");  scanf(" %c", &n);  char ch = k;  //First we must display our currently empty 32 bit result.  printf("\nResults before shifting and replacing the last 8 bits with %c\n", k);  display(result);  //we add our first character to the unsigned results and displays the value  result = packCharacters(0, k);  printf("\nResults after shifting and replacing the last 8 bits with %c\n", k);  display(result);  //next we will shift our next character onto the unsigned results.  //we send our previous version of our binary results and our current char  //to "push it" on the the binary value.  printf("\nResults before shifting and replacing the last 8 bits with %c\n", l);  display(result);  printf("\nResults after shifting and replacing the last 8 bits with %c\n", l);  result1 = packCharacters(result, l);  display(result1);  printf("\nResults before shifting and replacing the last 8 bits with %c\n", m);  display(result1);  //adds our next character  printf("\nResults after shifting and replacing the last 8 bits with %c\n", m);  result2 = packCharacters(result1, m);  display(result2);  printf("\nResults before shifting and replacing the last 8 bits with %c\n", n);  display(result2);  //adds and displays our final character.  printf("\nResults after shifting and replacing the last 8 bits with %c\n", n);  result3 = packCharacters(result2, n);  display(result3);  return 0;  }  //This function takes our unsigned version of our results and our next character  //and packs them onto the unsigned binary value by shifting the bits.  unsigned packCharacters(unsigned ch1, char ch2){  unsigned unsignedPack = ch1;  unsignedPack <<= 8;  unsignedPack |= ch2;  return unsignedPack;  }  //will display our 32 bit unsigned value as we create new versions of the  //results as we call it.  void display(unsigned result){  unsigned bitCount;  unsigned temp = 1 << 31;  //prints the unsigned value of the results.  printf("%7u = ", result);  for(bitCount=1; bitCount<=32; bitCount++){  result & temp ? putchar('1'): putchar('0');  result <<= 1;  if(bitCount % 8 == 0){  putchar(' ');  }  }  putchar('\n');  } |
|  |
| Enter first character: K  Enter second character: L  Enter third character: M  Enter fourth character: N  Results before shifting and replacing the last 8 bits with K  0 = 00000000 00000000 00000000 00000000  Results after shifting and replacing the last 8 bits with K  75 = 00000000 00000000 00000000 01001011  Results before shifting and replacing the last 8 bits with L  75 = 00000000 00000000 00000000 01001011  Results after shifting and replacing the last 8 bits with L  19276 = 00000000 00000000 01001011 01001100  Results before shifting and replacing the last 8 bits with M  19276 = 00000000 00000000 01001011 01001100  Results after shifting and replacing the last 8 bits with M  4934733 = 00000000 01001011 01001100 01001101  Results before shifting and replacing the last 8 bits with N  4934733 = 00000000 01001011 01001100 01001101  Results after shifting and replacing the last 8 bits with N  1263291726 = 01001011 01001100 01001101 01001110  Process returned 0 (0x0) execution time : 8.185 s  Press any key to continue.  Enter first character: H  Enter second character: I  Enter third character: J  Enter fourth character: K  Results before shifting and replacing the last 8 bits with H  0 = 00000000 00000000 00000000 00000000  Results after shifting and replacing the last 8 bits with H  72 = 00000000 00000000 00000000 01001000  Results before shifting and replacing the last 8 bits with I  72 = 00000000 00000000 00000000 01001000  Results after shifting and replacing the last 8 bits with I  18505 = 00000000 00000000 01001000 01001001  Results before shifting and replacing the last 8 bits with J  18505 = 00000000 00000000 01001000 01001001  Results after shifting and replacing the last 8 bits with J  4737354 = 00000000 01001000 01001001 01001010  Results before shifting and replacing the last 8 bits with K  4737354 = 00000000 01001000 01001001 01001010  Results after shifting and replacing the last 8 bits with K  1212762699 = 01001000 01001001 01001010 01001011  Process returned 0 (0x0) execution time : 4.638 s  Press any key to continue. |