Assignment 9

7.9)

a. unsigned int values[SIZE] = {2,4,6,8,10};

b. unsinged int \*vPtr;

c. for(i = 0; i < SIZE; i++)

printf(“%d “, values[i]);

d. vPtr = values;

vPtr = &values[0];

e. for(I = 0; I < SIZE; i++)

printf(“%d “, \*(vPtr + 1));

f. for(I = 0; I < SIZE; i++)

printf(“%d “, \*(values + 1));

g. for(I = 0; I < SIZE; i++)

printf(“%d\n”, vPtr[i]);

h. values[4], vPtr[4], \*(vPtr + 4), \*(values + 4)

i. 1002508 and 8

j. 1002500 and 2

7.10)

Value1 and value 2 are long int.

Value1 memory = 200000

a. long \*1Ptr;

b. 1Ptr = &value1;

c. printf(“%ld\n”, \*1Ptr);

d. \*1Ptr = value2

e. printf(“%ld”, value2);

f. printf(“%p”, &value1);

g. printf(“%p”, 1Ptr);

Both are the same since we only changed the value of the element, not the address.

7.11)

a. void zero(long int bigIntegers[])

b. void zero(long int[])

c. int add1AndSum(int oneTooSmall[])

d. int add1AndSum(int[])

7.21) a, c, e, g

a. Change %d to %p to get the memory location and replace \* with & from number in printf.

Correct: This will return the memory address

Int \*number;

printf(“%p\n”, &number);

or remove \* from number.

Other:

Int number;

Printf(“%d\n”, number);

c. add ‘&’ in front of y to assign its address to X.

correct:

int \* x, y;

x = &y;

e. Change printf statement, change void to short, and remove \* from line 3.

Correct

short \*numPtr, result;

short \*genericPtr = numPtr;

result = (genericPtr + 7);

g. Change the printf statement to print the address.

Correct:

Char \*s;

Printf(“%p\n”, &s);

Chapter 9:

9.4)

a. printf(“%-15.8u\n”, (unsigned)40000);

c. printf(“With a sign: %+d\nWithout a sign: %d\n”, 200, 200);

e. while (input != 'p'){

scanf("%c", &input);

if(input == 'p'){

break;

}else{

s[I] = input;

}

}

I++;

}

g. scanf(“%d%\*c%d%\*c%d”, &hour, &minute, &second);

i. scanf(“%d:%d:%d”, &hour, &minute, &second);

9.5)

a. “10000 “ (5 blank spaces)

c. 1024.988

e. “ 1000000

+1000000”

g. “ 4.45E+02” (2 spaces before number)

9.6)

a. remove single quotes

correct:

printf(“%s\n”, “Happy Birthday”);

c. change “c” to “s”

correct:

printf(“%s\n”, “This is a string”);

e. add “&” in front of the day array in printf

correct:

char day[] = “Sunday”;

printf(“%s\n”, &day[3]);

g. add quotes around “%f”

correct:

printf(“%f”, 123.456);

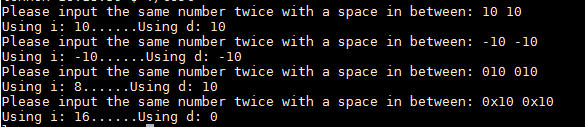
i. add “&” in front of the array since its scanf.

correct:

char s[10];

scanf(“%c”, &s[7]);

9.7)



Since “%d expects an integer and “0x10” is hex, it will not handle it properly, so we get 0.

#include <stdio.h>

int main()

{

int x = 0, y;

while(x != -1){

printf("Please input the same number twice with a space in between: ");

scanf("%i%d", &x, &y);

printf("Using i: %i......Using d: %d\n", x, y);

}

return 0;

}

7.30) (Extra Credit)