1. Which of the following statements correctly defines ClassNum as a constant value?

i. #constant ClassNum 16.216

ii. #define ClassNum 16.216

iii. #define ClassNum = 16.216

iv. double ClassNum = 16.216;

2. Which of the statements below has the same result as y = x \* 2?

i. y = x >> 1;

ii. y = x << 1;

iii. y = x >> 2;

iv. y = x << 2;

v. None of the above

c. Given an integer val, which of the following if statements prints val only if val is a positive, non-zero value that is less than 100?

i. if ((val != 0) || (val < 100))

printf(“Value = %d\n”, val);

ii. if ((val != 0) && (val < 100))

printf(“Value = %d\n”, val);

iii. if ((val > 0) || (val < 100))

printf(“Value = %d\n”, val);

iv. if ((val > 0) && (val < 100))

printf(“Value = %d\n”, val);

v. None of the above

a. You are given the following short program:

int main() {

int i;

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

if (i == 0)

continue;

printf("i = %d\n", i);

}

printf("Loop done\n");

}

What is the first line of output this program prints?

i. i = 0

ii. i = 1

iii. Loop done

iv. Program prints no output—it exits before it prints anything.

{

x = x – 10;

i++;

}

ii. i = x;

while (i > 60) {

i = i – 10;

}

iii. i = 0;

while (i < 8) {

x += (-10);

i += 2;

}

iv. i = 5;

while (i >= 1) {

x = x – 10;

i--;

}

d. Say we have a function, declared as follows:

void foo(int \*x, int \*y);

If your program contains two integers, a and b, which of the choices below correctly calls foo and passes the addresses of a and b to that function?

i. foo(a,b);

ii. foo(\*a,\*b);

iii. int \*ptr = &a;

foo(\*ptr, b);

iv. int \*ptr = &a;

foo(ptr, &b);

The following question uses the structure defined below:

typedef struct {

char name[50];

double progs[8];

double exams[3];

} CourseData;

c. Which of the following choices is not a valid access to a field within a variable of type CourseData?

i. CourseData cd;

cd.progs[7] = 95;

ii. CourseData list[85];

list[0]->exams[0] = 83;

iii. CourseData s1;

scanf("%s", s1.name);

iv. typedef struct {

CourseData students[49];

} SectionData;

SectionData sd;

sd.students[0].exams[1] = 75;

int x = 5;

int y = 10;

if ((x % 2) == 1)

y = y + x;

x = x + 1;

What are the values of x and y at the end of the sequence?

i. x == 5, y == 10

ii. x == 6, y == 10

iii. x == 5, y == 15

iv. x == 6, y == 15

Say you have the following conditional statement:

if (a < 0) {

x = x + 1;

}

else if (b >= 0) {

x = x – 1;

}

else {

x = 0;

}

Assume x == 5 initially. Which values of a and b shown below would cause x to be set to 0 when the code above is executed?

i. a == -2, b == -2

ii. a == 1, b == -1

iii. a == 2, b == 5

iv. a == 0, b == 0

a. You are given the following short program:

int i;

for (i = 0; i < 20; i++) {

if (i < 10)

continue;

if ((i % 2) == 1)

break;

}

How many iterations will this loop execute?

i. 10

ii. 11

iii. 12

iv. 20

v. An infinite number—the loop never ends

b.You aregiven the following short program:

void main() {int x, y;int \*p1, \*p2;

scanf("%d", &x);y = x;p1 = &x;p2 = p1;

**x = x + 2;// REPLACE THIS STATEMENT**

printf("%d", x);}

Which of the following statements could replace the underlined statement andalwaysgenerate the exact same program output?

i.y = y + 2;

ii.p1 = p1 + 2;

iii.\*p1 = \*p2 + 2;

iv.None of the above

void main() {

int d1, d2;

int \*p1, \*p2;

d1 = 16;

d2 = 216;

p1 = &d2;

p2 = &d1;

d2 = \*p2 - 15;

\*p2 = \*p1 + 2;

p1 = p2;

d1 = d2 \* 4;

\*p1 = \*p2 + 15;

printf("%d %d %d %d\n", d1, d2, \*p1, \*p2);

}

The following question uses the structure defined below:

typedef struct {

int number;

char name[40];

char rating[7];

int length;

char time[4][7];

} Movie;

d. Which of the following choices is not a valid access to a field within a variable of type Movie?

i. Movie m;

scanf("%s", m.rating);

ii. typedef struct {

Movie mList[10];

} TheaterData;

TheaterData td;

td.mList[0]->length = 120;

iii. Movie m;

m.name[0] = 'A';

iv. Movie list[100];

strcpy(list[5].name, "Dude, Where's My Car?");

/////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////

读程序，写结果：

int main() {

int v1, v2;

int \*p1, \*p2;

v1 = v2 = 0;

p1 = &v2;

p2 = p1;

v2 = 16;

\*p2 = 24;

v1 = \*p1 + 16;

\*p1 = \*p2 – 10;

printf("%d %d\n", v1, v2);

printf("%d %d\n", \*p1, \*p2);

return 0;

}

double f(double x, double y) {

y \*= 2;

x -= 3;

return (x + y) / 2.0;

}

int main() {

double q, r, s;

q = f(5, 8);

r = f(8, 5);

s = f(q, r);

printf("%.2lf %.2lf %.2lf\n", q, r, s);

return 0;

}

int f1(int \*arg1) {

(\*arg1)++;

return (\*arg1) \* 2;

}

int f2(int arg2) {

return f1(&arg2) + 10;

}

int f3(int \*arg3) {

return f1(arg3) + 10;

}

int main() {

int a, b, c;

int x, y, z;

a = b = c = 10; // Set all three values to 10

x = f1(&a);

y = f2(b);

z = f3(&c);

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

printf("%d %d %d\n", x, y, z);

return 0;

}

int main() {

int x[10] = {0};

int \*ptr = x;

int i;

ptr[2] = 16;

ptr[1] = 216;

ptr[6] = 2011;

ptr = ptr + 5;

ptr[0] = 12;

ptr[1] = 20;

ptr[-2] = 3;

ptr = ptr - 2;

ptr[4] = ptr[5] = ptr[6] = 201;

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

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

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

return 0;

}

int main() {

char str[] = "Applications";

char \*ptr1, \*ptr2;

char temp;

ptr1 = str;

ptr2 = &str[11];

while (\*ptr1 != '\0') {

temp = \*ptr1;

\*ptr1 = \*ptr2;

\*ptr2 = temp;

ptr1 += 4;

ptr2 -= 3;

}

printf("%s\n", str);

return 0;

}

int main() {

int orig[] = {75, 82, 13, 90, 51, 58, -1};

int scaled[6];

int \*po = orig;

int \*ps = scaled;

int i;

while (\*po != -1) {

\*ps = \*po + 6;

po++;

ps++;

}

for (i = 0; i < 6; i++) {

po--;

ps--;

printf("%d %d\n", \*po, \*ps);

}

return 0;

}

void badSwap(int \*x, int y) {

int temp;

temp = \*x;

\*x = y;

y = temp;

}

void badSwap\_2(int x, int \*y) {

int temp;

temp = x;

x = \*y;

\*y = temp;

}

void main() {

int a = 1;

int b = 2;

int c = 3;

int d = 4;

badSwap(&a, b);

badSwap\_2(c, &d);

badSwap(&b, c);

badSwap\_2(a, &d);

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

}

void main() {

char str[] = "Show sample string\n";

int v[] = {6, 10, -11, -2, 7, 4, 4};

int i;

char \*p = str;

for (i = 0; i < 7; i++) {

p = p + v[i];

printf("%c", \*p);

}

}

void main() {

char str1[20];

char str2[30];

int n;

strcpy(str1, "16.216");

strncpy(str2, "Spring 2012 Section 201", 11);

str2[11] = '\0';

printf("%s %s\n", str1, str2);

n = strlen(str1);

printf("str2[%d] = %c\n", n, str2[n]);

strncat(str2, str1, 4);

strncat(str1, str2, 4);

printf("%s\n%s\n", str1, str2);

}

void main() {

char s1[] = "AbCdEfG";

char s2[] = "AbCDEfg";

int i = 1;

while (i <= strlen(s1)) {

if (strncmp(s1, s2, i) != 0) {

printf("%d: N\n", i);

}

else

printf("%d: Y\n", i);

i++;

}

}

//////////////////////////////////////////////////////

// 编程

c. int removeSub(char str[], int start, int end);

 Given: a string str, and positions within that string, start and end.

o You can assume start and end do not need to be checked for errors.

o Neither position references the null character at the end of str.

 Remove characters from str between positions start and end, including those two positions, and shift the remaining characters so you now have a shorter string.

 Return the number of characters in the new string.

 Examples: Assume you have an array char s[] = "Test string";

o removeSub(s, 0, 0) returns 10; s = "est string"

o removeSub(s, 0, 4) returns 6; s = "string"

o removeSub(s, 4, 7) returns 7; s = "Testing"

int removeSub(char str[], int start, int end) {

int pos1, pos2; // Positions within string str[]