**90 Minutes** **CENG 230 Final Exam** **14.01.2014 – 13:30**

There are 50 questions (each 2 points) for a total of 100 points. Exam Type: **A**

**Do not forget to code your student ID and exam type correctly to the answer sheet. Otherwise, your answers will not be graded.**

All questions are multiple choice, **no points will be lost for wrong answers.**

**Name:** **Student ID:** **Section:** **Signature:**

A A A A A A A A A A A A A A A A A A A A A A A A A A A A A

**1) Which of the following is not a valid variable name?**

**a)** foo **b)** _foo **c)** foo123 **d)** 123foo **e)** foo_

**2) If *a* is set to be 5, *b* is set to be 2, *c* is set to be 1; what is the result of following arithmetic expression?**

```
a%5+b*a/2-c+(a+b)
```

**a)** 5 **b)** 4 **c)** 20 **d)** 11 **e)** 13

**3) If *a* is 5 and *b* is 7 what is the value of *a* and *b* after the execution of the given code?**

```
if(a=1) { a+=3;  b--; }
else if(b==6) {a=5; b=10; }
else{ a=7; b=a+b; }
```

**a)** 8,6 **b)** 5,10 **c)** 7,14 **d)** 7,13 **e)** 4,6

**4) If *a* is 5, *b* is 4 , *c* is 10 what is the output?**

```
a=b=c+6%2;
printf("%d %d %d", a,b,c);
```

**a)** 5 10 10 **b)** 13 13 10 **c)** 10 10 10 **d)** 5 4 10 **e)** 13 10 10

**5) What is the output of the below code segment ?**

```
int i=32;
char c;
c=i;
printf("%d", c);
```

**a)** 23 **b)** 'c' **c)** 69 **d)** 'E' **e)** 32

**6) How can you code the below condition in C after including math.h library?**

if log(*x) is greater than y then print "x>y"* else *print "x<y"*

**a)** `log(x)>y?print("x>y"):print("x<y");`
**b)** `ln(x)>y?printf("x>y"):printf("x<y");`
**c)** `if (log(x)>y) printf("x>y");else printf("x<y");`
**d)** `(log x )>y?print("x>y"):print("x<y");`
**e)** `log10(x)>y?printf("x>y"):printf("x<y");`

**7) What is the output of the below code segment after including math.h library?**

```
double x=4;
printf("%d\n", (int)pow(sqrt(x),2));
```

**a)** 2 **b)** 4 **c)** 4.00 **d)** 2.00 **e)** 4.000000

**8) What is the output of the below code segment?**

***Note: The ascii code of character A is 65.***

```
char cresult, c1 = 88, c2 = 3, c3 = 4;
cresult = c1 * c2 / c3;
printf("%c\n",cresult);
```

**a)** b **b)** 'A' **c)** 66 **d)** B **e)** A

**9) What will the following program print?**

```
float f=3.14;
int i;
i=f;
printf("%d\n", i);
```

**a)** 3.14 **b)** 3.140000 **c)** 3 **d)** 3.000000 **e)** 0

**10) What will be the output of the following code segment?**

```
int a=1,b=2,c=10;
while (b<c){
a+=--c-b++;
printf("%d ",a);  }
```

**a)** 7 12 15 16 **b)** 8 13 16 17 **c)** 8 12 15 17 **d)** 6 11 14 15 **e)** 7 11 14 16

A A A A A A A A A A A A A A A A A A A A A A A A A A A A A

**11) What should be the values of a, b, c, d in below expression in order for the final value of x to be 1 ?**

```
int x = -1;
if(a==0)
     if(b>3)
        if(c<4) x=4;
        else if (d=1) x=3;
        else {
                if ((b-c) && --d ) x=1;}
     else x=0;
```

a) `a=0, b>3, c>4, d !=1, b-c !=0`
**b)** `a=0, b>3, c<4, d !=1, b-c !=0`
**c)** `a=0, b>3, c<4, d =1, b-c !=0`
**d)** `a=0, b>3, c>4, d !=1, b-c !=0`
**e)** `x will not be equal to 1`

**12) What will be the output of the following code segment?**

```
int x=4,y=2,z=3;
if(x<y);
if(x<z) printf ("Minimum is x ");
else printf ("Minimum is z ");
if(x>y);
if(y<z) printf ("Minimum is y ");
else printf ("Minimum is z ");
```

**a)** `Minimum is x` **b)** `Minimum is z` **c)** `Minimum is y`
**d)** `Minimum is z Minimum is z`
**e)** `Minimum is z Minimum is y`

**Use below code segment for the next two questions.**

```
int a,b,c;
scanf("%d %d %d",&a,&b,&c);
switch(a%b){
case 0: printf("%d",a-b);
case 1: printf("%d",b-c);break; }
printf("%d",b-c);
```

**13) What is the output, if the input is `6 2 1` ?**

**a)** 11 **b)** 411 **c)** 421 **d)** 111 **e)** 121

**14) What is the output, if the input is `5 9 4` ?**

**a)** 5 **b)** 5-5 **c)** -455 **d)** -5 **e)** -4-5

**15) What will be the output of the following code segment?**

```
int a=1,b=3,c=1;
if ((a||c--)&&(c&&b--)) printf("%d",b);
printf("%d %d %d ",a,b,c);
```

**a)** 1 2 0 **b)** 21 2 0 **c)** 21 2 1 **d)** 1 3 1 **e)** 31 3 1

**16) What will be the output of the following code segment?**

```
printf("%d",(3/5*2+3||3*2+6/7&&5-8/2+1)-3);
```

**a)** 1 **b)** 0 **c)** -1 **d)** -2 **e)** -3

**17) Which one of the following statement is wrong?**

**a)** Logical operators have less precedence than relational operators.
**b)** '&&' operator have more precedence than '||' operator.
**c)** Ternary conditional operator has left-to-right associativity.
**d)** '&&' and '||' operators always have short-circuit evaluation
**e)** '*', '/', '%', addition and subtraction operations are done before the relational operations.

A A A A A A A A A A A A A A A A A A A A A A A A A A A A

**18) What would be the output of the following code portion?**

```
int i=3, j=4, k=8;
for ( ; ; i++, j++, k--) {
   if (j%2) continue;
   if (i>k) break;  }
printf("%d %d %d", i, j, k);
```

**a)** 6 6 4 **b)** 6 8 5 **c)** 7 8 5 **d)** 7 8 4 **e)** 8 8 5

**19) What would be the output of the following code portion?**

```
int i, j, sum;
for (sum=i=0, j=10; i<j; i++, j--)
  sum += i + j;
printf("%d %d %d", i, j, sum);
```

**a)** 5 5 10 **b)** 5 5 50 **c)** 5 5 100 **d)** 6 4 10 **e)** 6 4 110

**20) What would be the output of the following code portion?**

```
int i=0, j=10, n=0;
do {
  n += i;
} while (i++ < j--);
printf("%d %d %d", i, j, n);
```

**a)** 5 5 5 **b)** 5 5 6 **c)** 5 5 15 **d)** 6 4 6 **e)** 6 4 15

**21) Assuming the user input will be `1 3 5 7 -1 -3` what would be the output of the following code portion?**

```
int count=0, sum=0, n;
do {
 scanf("%d", &n);
 count++;
 sum += n;
} while (n > 0);
printf("%d %d", count, sum);
```

**a)** 4 16 **b)** 4 15 **c)** 5 16 **d)** 5 15 **e)** 6 12

**22) What would be the output of the following code portion?**

```
int i, j, k, sum;
for (sum=i=0; i<4; i++)
for (j=i; j<4; j++)
for (k=j; k<4; k++)
  sum++;
printf("%d", sum);
```

**a)** 4 **b)** 12 **c)** 16 **d)** 20 **e)** 64

**23) What would be the output of the following code portion?**

```
int i=0, j, n=0;
while (i < 5) {
    j = 5;
    while (j > 0) {
       j--;
       if (j>i) continue; n++; }
    i++;  }
printf("%d", n);
```

**a)** 0 **b)** 5 **c)** 10 **d)** 15 **e)** Nothing (infinite loop)

**Use the following code portion for the next two questions.**

```
int i, j, k, sum=0;
for (i=0; i<3; i++){
j = 0;
while (j <= 3) {
k = j++;
do {
sum++;
} while (k++ <= 3);   }  }
```

**24) What would be the value of the variable sum after the above code executes?**

**a)** 9 **b)** 27 **c)** 36 **d)** 42 **e)** 81

**25) What would be the values of the variables i, j, k after the above code executes?**

**a)** 3 3 3 **b)** 3 4 4 **c)** 3 4 5 **d)** 4 4 4 **e)** 4 4 5

A A A A A A A A A A A A A A A A A A A A A A A A A A A A

**26) What is the output?**

```
# include <stdio.h>
void f1(int y)
{     int x= 2;
      printf("%d",x+y); x++;  }
int main (void)
{    int x=7,y=2;
     printf("%d",x); f1(x);
     printf("%d",x);
     return 0;}
```

**a)** 737 **b)** 3797 **c)** 797 **d)** 297 **e)** 237

**27) What is the output?**

```
int y=5,x=2;
void f1 (int *p)
{ *p=*p+y;}
int main (void)
{ int x[5]={1,2,3,4,5},y=3;
printf("%d,", *x);
f1(x);
printf("%d", *x);
return 0;}
```

**a)** 1,1 **b)** 2,2 **c)** 2,7 **d)** 1,6 **e)** 230,2

**28) Given the function definition statement as:**

`float myfun (int b[],int *a,double c,double d)`

**which of the below is the prototype of the function myfun?**

**a)** `float  myfun(int *,int *,double,double);`
**b)** `void myfun (int[],int *,double,double);`
**c)** `float myfun (int &,int *,double*2);`
**d)** `double myfun (2*int*2,double*2);`
**e)** `float myfun (int [],int *,2*double);`

**29) What is the output?**

```
int f1 (int x)
{ int y=2;
   printf("%d%d",x,y);
   return x++;
  return ++y; }
int main (void)
{ int y=5, x=5;
  printf("%d%d\n",f1(y),y);
  return 0;}
```

**a)** 25424 **b)** 5255 **c)** 5555 **d)** 5256 **e)** 5552

**30) What is the output?**

```
int main (void)
{ int x[5]={1,2,3,4,5},p;
   printf("%d,", f1(x[0],&x[3]));
   p=f1(x[0],&x[3]);
   printf("%d,%d,%d", p,x[0],x[3]);
  return 0; }
int f1 (int p, int*q)
{ *q=*q+5;
   p=*q+5;
   return p;}
```

**a)** 14,19,1,14 **b)** 14,14,14,14 **c)** 14,19,14,14 **d)** 1,19,1,14 **e)** 1,19,14,14

**31) What is the output?**

```
void f1 (void)
{ int y=5;
printf("%d",y); y++;
printf("%d",y);}
int main (void)
{ int y=3;
printf("%d",y);
f1();
printf("%d",y); return 0;}
```

a) 3563 b) 563563 c) 563566 d) 3566 e)3567

A A A A A A A A A A A A A A A A A A A A A A A A A A A

**32) What is the output?**

```
void f1 (int x)
     { int y=2;
       printf("%d%d",x,y);
       x++; }
int main (void)
     { int y=5, x=5;
       printf("%d%d",x,y);
       f1(y);
       printf("%d",x);
       return 0;}
```

**a)** 55256 **b)** 255255 **c)** 55526 **d)** 255256 **e)** 55525

**33) If a function modifies a variable defined in the main function, then this is called:**

**a)** Side Effect or Call by Reference
**b)** Call by Value
**c)** "address of" operator (&)
**d)** "Content of" operator (*)
**e)** Prototype and/or Function Definition Statement

**34) To prevent a global variable from interfering with execution of a function;**

a) All the variables used in main funcion must be declared.
b) All the variables used in the function should be declared.
c) All the functions must be linked to main by using #include directive.
d) Function(s) must be coded before main.
e) Function(s) must be coded after main.

**35) What is the output?**

```
 int i,a[]={1,2,3,4,5},b[]={10,20,30,40,50};
for(i=1; i<5; i++)
   b[i]=a[i]+b[i-1];
for(i=0; i<5; i++)
   printf("%d ",b[i]);
```

**a)** 10 12 15 19 24 **b)** 1 12 23 34 45 **c)** 11 22 33 44 55
**d)** 10 32 53 74 95 **e)** 21 32 43 54 50

**36) What is the output?**

```
int i,a3[4];
  for(i=0; i<4; i++)
     a3[i]=i*2+1;
  for(i=0; i<3; i++)
     a3[i]=a3[i+1];
  a3[3]=a3[0];
  for(i=0; i<4; i++)
     printf("%d",a3[i]);
```

**a)** 1111 **b)** 3571 **c)** 3573 **d)** 5793 **e)** 7777

**37) What is the output?**

```
 int ar1[]={1,2,3,4,5};
  int ar2[3]={9};
  printf("%d,%d",ar1[1],ar2[1]);
```

**a)** 0,9 **b)** 1,9 **c)** 1,0 **d)** 2,0 **e)** 2,9

**38) What is the output?**

```
      int a[5]={11,1,16,-1,13};
      printf("%d",a[a[2]-a[4]]);
```

**a)** -1 **b)** 1 **c)** 2 **d)** 11 **e)** 16

**39) What is the output?**

```
      int j,c,x[]={5,-3,-1,7,8,-2,0,9,-6,8};
      for(c=0,j=0;j<10;j++)
      if (x[j]<0) c=c+j;
      printf("%d",c);
```

**a)** 0 **b)** 16 **c)** 20 **d)** 8 **e)** 4

**40) Which one of the following declarations is wrong and causes a compile error?**

**a)** `char a[]={'a',61,'9'};` **b)** `int b[5]={0,1};`
**c)** `float c[]={5.5,3};` **d)** `int d[]={};`
**e)** `double e[3]={7.8,1.0,3.5};`

A A A A A A A A A A A A A A A A A A A A A A A A A A A

**41) What is the output of this program.**

```
   #include <stdio.h>
   void f(int a, int *b){
      a = a + 5;
      *b = a; }
   void main(){
      int arr[2]={10,20};
      f(arr[0],&arr[1]);
      printf("%d,%d",arr[0],arr[1]); }
```

**a)** 10,10 **b)** 10,15 **c)** 10,20 **d)** 15,15 **e)** 15,20

**42) What is the output of this program.**

```
#include <stdio.h>
int g(int* a1, int* a2){
   *a1 = *a2 + 5;
   *a2 = *a2 * 2;
   return *a2 - *a1;   }
void main(){
   int a[3]={10,20,30};
   a[2] = g(&a[0],a+1);
 printf("%d,%d,%d",a[0],a[1],a[2]); }
```

**a)** 26,42,16 **b)** 16,20,6 **c)** 25,40,10 **d)** 15,40,35 **e)** 25,40,15

**43) What is the output?**

```
void main() {
  int A[10][10]={{1,2,3},{3,4,5}};
  int B[10][10]={{2,2,2},{5,4,3}},C[10][10];
  int i,j, N=2, M=3;
  for (i=0;i<N;i++)
    for (j=0;j<M;j++)
      if (A[i][j]>B[i][j])
       C[i][j]=A[i][j];
      else
      C[i][j]=B[i][j];
  for (i=0;i<N;i++)
  {
    for (j=0;j<M;j++)
    printf("%d ",C[i][j]);
    printf("\n");
  }
}
```

**a)**
1 2 3
3 4 5

**b)**
2 5
2 4
2 3

**c)**
1 2 3
5 4 3

**d)**
2 5
2 4
3 5

**e)**
2 2 3
5 4 5

**44) What is the output?**

```
 void main() {
  int A[10][10]={{1,2,3},{3,4,5}};
  int B[10][10]={{2,2,2},{5,4,3}},C[10][10];
  int i,j, N=2, M=3;
  for (i=0;i<N;i++)
    for (j=0;j<M;j++)
      if (A[i][j]>B[i][j])
       C[j][i]=A[i][j];
      else
      C[j][i]=B[i][j];
  for (i=0;i<M;i++)
  {
    for (j=0;j<N;j++)
    printf("%d ",C[i][j]);
    printf("\n");
  }
}
```

**a)**
1 2 3
3 4 5

**b)**
2 5
2 4
2 3

**c)**
1 2 3
5 4 3

**d)**
2 5
2 4
3 5

**e)**
2 2 3
5 4 5

A A A A A A A A A A A A A A A A A A A A A A A A A A A A

**45) What is the output?**

```
#include <stdio.h>
 void f3(int n, int a[]) {
   int i;
    for (i=0;i<n;i++)
      a[i]++;
    for (i=0;i<n;i++)
      printf("%d ",a[i]);
    printf("\n"); }
void main() {
  int A[]={1,2,3,3,4,5},B[10];
  int i, N=6;
  for (i=0;i<N;i++)
    B[i]=A[i];
  f3(N,A);
  for (i=0;i<N;i++)
     printf("%d ",A[i]);
  printf("\n");
  for (i=0;i<N;i++)
     printf("%d ",B[i]);
  printf("\n");  }
```

**a)** 1 2 3 3 4 5 1 2 3 3 4 5 1 2 3 3 4 5 **b)** 2 3 4 4 5 6 2 3 4 4 5 6 1 2 3 3 4 5 **c)** 1 2 3 3 4 5 2 3 4 4 5 6 1 2 3 3 4 5 **d)** 2 3 4 4 5 6 1 2 3 3 4 5 1 2 3 3 4 5 **e)** 2 3 4 4 5 6 2 3 4 4 5 6 2 3 4 4 5 6

**46) What is the output?**

```
#include <stdio.h>
void f4(int n, int a[])
{
   int i; int b[10];
   for (i=0;i<n;i++)
      b[i]=a[i];
   for (i=0;i<n;i++)
    b[i]++;
   for (i=0;i<n;i++)
     printf("%d ",a[i]);
   printf("\n");
   for (i=0;i<n;i++)
     printf("%d ",b[i]);
  printf("\n");
}
void main() {
  int A[]={1,2,3,3,4,5};
  int i, N=6;
  f4(N,A);
  for (i=0;i<N;i++)
     printf("%d ",A[i]);
  printf("\n");
}
```

**a)** 1 2 3 3 4 5 1 2 3 3 4 5 1 2 3 3 4 5 **b)** 2 3 4 4 5 6 2 3 4 4 5 6 1 2 3 3 4 5 **c)** 1 2 3 3 4 5 2 3 4 4 5 6 1 2 3 3 4 5 **d)** 2 3 4 4 5 6 1 2 3 3 4 5 1 2 3 3 4 5 **e)** 2 3 4 4 5 6 2 3 4 4 5 6 2 3 4 4 5 6

**47) What is the output?**

```
#include <stdio.h>
void main () {
 int A[10][10]={{1,2,3},{3,4,5},{5,6,7}};
 int i,j,N=3,t;
 for (i=0;i<N;i++)
 {
  t=A[i][i];
  A[i][i]=A[i][N-i-1];
  A[i][N-i-1]=t;
 }
 for (i=0;i<N;i++)
 {
  for (j=0;j<N;j++)
  printf("%d ",A[i][j]);
  printf("\n");
 }  }
```

**a)** 1 2 3 / 3 4 5 / 5 6 7 **b)** 5 2 7 / 3 4 5 / 1 6 3 **c)** 1 2 3 / 5 4 3 / 5 6 7 **d)** 1 6 3 / 3 4 5 / 5 2 7 **e)** 3 2 1 / 3 4 5 / 7 6 5

A A A A A A A A A A A A A A A A A A A A A A A A A A A A

**48) What is the output?**

```
#include <stdio.h>
void main() {
   char a[]="abc";
   char b[]={'a','b','c','\0'};
   char c[]={'a','b','c',0};
   char d[]={'a','b','c'};
printf("%d %d %d\n",strlen(a),strlen(b),strlen(c));
   printf("%d\n",strlen(strcpy(d,"\0")));
   printf("%s\n",d);
}
```

**a)** 3 3 3 / 0 **b)** 3 4 4 / 4 / abc **c)** 4 4 4 / 4 / abc **d)** 3 3 4 / 3 / abc **e)** 3 4 3 / 4 / abc

**49) What is the output?**

```
 #include <stdio.h>
 void main () {
   char e[10],f[10];
   e[0]='a';e[1]='b';e[2]='\0';
   strcat(e,"c");
   printf("%d\n",strlen(e));
   printf("%s\n",e);
   strcpy(f,e);
   f[2]='d';
   printf("%d\n",strlen(f));
   printf("%s\n",f);
}
```

**a)** 3 / abc / 3 / adc **b)** 3 / abc / 3 / abd **c)** 2 / ab / 2 / ab **d)** 3 / abc / 4 / abcd **e)** 3 / abc / 2 / ad

**50) What is the output?**

```
   #include <stdio.h>
void main () {
 char e[10],f[10];
 strcpy(e,"abc");
 strcpy(f,"abd");
 if (strcmp(e,f))
 printf("%s\n",e);
else
printf("%s\n",f);
 f[3]='x';f[4]='\0';
 printf("%s\n",strchr(f,'d'));
 printf("%s\n",strstr(f,"d"));
}
```

**a)** abc / d / d **b)** abd / dx / dx **c)** abc / dx / dx **d)** c / d / d **e)** c / x / x