MCS 494 quiz
2004.08.30
D. J. Bernstein

Your answers must be based solely on your own knowledge and the information on these sheets. You are not permitted to use books, notes, or computers. Do not ask your proctor for interpretations or clarifications.

Do not hand in these sheets. Anything that you want to have graded must appear in the answer booklet. Make sure that your name is on the front of the booklet.

The question in each case is "What does this program print?" Write your answer in the same format that it will appear on the screen, preceded by the problem number and a dot. Assume that none of the programs run out of memory.

Problem 1.

```
char x[1000] = "Hello";
int main(void) { printf("%d\n",strlen(x)); return 0; }
```

Problem 2.

```
int main(void)
{ int u = 6; int *p = &u;
  printf("%d\n",u); *p = *p + 1; printf("%d\n",u);
}
```

Problem 3.

```
int main(void)
{ int x[5] = { 8, 6, 7, 5, 3 }; int *p = x + 2; int i;
  for (i = 0;i < 5;++i) printf("%d\n",*p);
}
```

Problem 4.

```
struct point { int x; int y; } ;
struct point f[3] = { { 2, 7 }, { 1, 8 }, { 2, 8 } };
int i;
int main(void)
{ for (i = 0;i < 3;++i) printf("%d\n",f[i].y); return 0; }
```

Problem 5.

```
int main(void)
{ int n = 31415;
  printf("%d\n",sscanf("27182","%3d",&n));
  printf("%d\n",n);
}
```

Problem 6.

```
struct point { double x; double y; } ;
void rotatepoint(struct point p)
{ double oldx = p.x;
  p.x = -p.y;
  p.y = oldx;
}
int main(void)
{ struct point p = { 2.7, 1.8 };
  rotatepoint(p);
  printf("%f %f\n",p.x,p.y);
  return 0;
}
```

Problem 7.

```
int main(void)
{
  int i, j;
  for (i = 0;i < 5;++i) {
    for (j = i;j < 5;++j)
      printf("%d",j);
    printf("\n");
  }
  return 0;
}
```

Problem 8.

```
int main(void)
{
  int x[5];
  int i;
  for (i = 0;i < 5;++i) x[i] = i * i;
  for (i = 0;i < 3;++i) x[i] = x[i + 2];
  for (i = 0;i < 3;++i) x[i + 2] = x[i];
  for (i = 0;i < 5;++i) printf("%d\n",x[i]);
  return 0;
}
```

Problem 9.

```
int main()
{
  int n = 17;
  while (n > 1)
    switch (n % 2) {
      case 0:
        n /= 2;
        break;
      default:
        printf("%d\n",n);
        n *= 3;
        n += 1;
    }
  return 0;
}
```

Problem 10.

```
int x[5] = { 3, 1, 4, 1, 5 };
int i = 2;
int j = 10;
int k = 0;

int main(void)
{
  while (i < 4) x[i++] = (j += ++k);
  while (i > 0) printf("%d\n",x[i--]);
  return 0;
}
```

Problem 11.

```
int main(void)
{
  int i = 4;
  int j = 5;

  while (i--) if (i % 2)
  while (j--) printf("%d\n",j);
  else j = i; return 0;
}
```

Problem 12.

```
int main(void)
{ char *y = malloc(10);
  if (y == malloc(20)) printf("One\n");
  if (y == malloc(10)) printf("Two\n");
  if (y == y) printf("Three\n");
}
```

Problem 13.

```
int main(void)
{
  int x[] = { 3, 1, 4, 1, 5, 9, 2, 6 };
  int *p = x; int *q = x + 7; int i;
  while (p != q)
    if (*p < 4) ++p;
    else if (*q >= 4) --q;
    else { i = *p; *p = *q; *q = i; }
  for (i = 0;i < 8;++i) printf("%d\n",x[i]);
}
```

Problem 14.

```
int main(void)
{ char *q = "less fried food, more aerobic exercise";
  char *p = q;
  while (*p) if (*p++ == ’ ’) { printf("%d\n",p - q); q = p; }
}
```

Problem 15.

```
void doit(int n)
{ while (n > 0) { putchar("012345678!"[n % 10]); n /= 10; } }
int main(void)
{ doit(127); doit(31415926); }
```

Problem 16.

```
void doit(int n)
{ int i;
  for (i = 0;i < n;++i) doit(i);
  printf("%d\n",n);
}
int main(void) { doit(3); return 0; }
```

Problem 17.

```
struct i { int d; struct i *n; } ;
struct i x[5] = {
  { 3, x + 2 }
, { 1, x + 1 }
, { 4, x + 3 }
, { 1, x + 4 }
, { 5, 0 }
} ;
int main(void)
{ struct i *t = x;
  while (t) { printf("%d\n",t->d); t = t->n; }
  return 0;
}
```

Problem 18.

```
int x[8] = { 8, 7, 2, 9, 5, 4, 3, 1 };
int main(void)
{ int *p = x + 3; int *q = x + 6;
  while (p != x) {
    if ((q != x + 3) && (q[-1] < p[-1]))
      printf("%d\n",*--q);
    else
      printf("%d\n",*--p);
  }
  while (q != x)
    printf("%d\n",*--q);
  return 0;
}
```

Problem 19.

```
struct i {
  int d;
  struct i *n;
} x[12];

int main(void)
{ struct i *t = x + 8;
  while (t > x) {
    --t;
    t->d = t - x;
    t->n = t + 1;
  }
  t[7].n = t;
  while (t != t->n) {
    t = t->n;
    printf("%d\n",t->n->d + 1);
    t->n = t->n->n;
    t = t->n;
  }
  return 0;
}
```

Problem 20.

```
void doit(int p)
{ static char x[25] = "...X..X.X..X....XXXX.....";
  int q = 0;
  x[p] = ’*’;
  if ((p >= 5)      && x[p-5] == ’.’) { q = 1; doit(p-5); }
  if ((p < 20)      && x[p+5] == ’.’) { q = 1; doit(p+5); }
  if ((p % 5)       && x[p-1] == ’.’) { q = 1; doit(p-1); }
  if (((p + 1) % 5) && x[p+1] == ’.’) { q = 1; doit(p+1); }
  if (!q) {
    while (q < 25) {
      do putchar(x[q++]); while (q % 5);
      putchar(’\n’);
    }
    putchar(’\n’);
  }
  x[p] = ’.’;
}
int main(void) { doit(0); return 0; }
```