

Name: \_\_\_\_\_

NetID: \_\_\_\_\_

Answer all questions in the space provided. Write clearly and legibly, you will not get credit for illegible or incomprehensible answers. This is a closed book exam. However, each student is allowed to bring one page of notes to the exam. Print your name at the top of every page.

1. (5 points) List five data types used in C.
2. (5 points) List five keywords that are used to control the “flow” of the program in C.
3. (5 points) What is the operator %= used for? Write an expression that uses it and show another way to write the same expression.
4. (5 points) The command line arguments to the C main can be read by accessing the parameters of main, if main has the following signature: `int main(int argc, char *argv[]);` What is contained in `argv[0]`?
5. (5 points) Why is it important to always initialize your variables in a C program?

6. (10 points) What is the output of this program?

```
#include <stdio.h>

void main(void)
{
    int a = 12;

    if (a % 3 == 0)
    {
        printf("A\n");
        a+=3;
    }
    else
    {
        printf("B\n");
        a-=5;
    }
    printf("C\n");
    if (a % 5 != 0)
    {
        printf("D\n");
        a+=4;
    }
    else
    {
        printf("E\n");
        if (a < 3)
        {
            printf("F\n");
            a++;
        }
        else if (a > 7)
        {
            printf("G\n");
            a--;
        }
    }
    printf("%d\n",a);
}
```

7. (12 points) What is the output of this program?

```
#include <stdio.h>

int a=7;

int foo(int n)
{
    int b=3;
    a++;
    b++;
    n = n+a+b;
    printf("foo: a=%d, b=%d, n=%d\n", a, b, n);
    return n;
}

void main(void)
{
    int a, n;
    n = 2;
    a = foo(n);
    printf("main: n=%d, a=%d\n", n, a);

    a = foo(n);
    printf("main: n=%d, a=%d\n", n, a);
}
```

8. (12 points) What is the output of this program?

```
#include <stdio.h>

void main(void)
{
    unsigned char x = 37;

    unsigned char a = x << 3;
    unsigned char b = x >> 3;
    unsigned char c = x & 15;
    unsigned char d = x & 222;
    unsigned char e = x | 15;
    unsigned char f = x ^ 15;

    printf("a=%d, b=%d, c=%d, d=%d, e=%d, f=%d\n",
           a, b, c, d, e, f);
}
```

9. (10 points) What is the output of this program?

```
#include <stdio.h>

void main(void)
{
    char s[] = "XdXXoXnXe";
    char del = 'X';

    int sourceIndex = 0;
    int sinkIndex = 0;
    while (s[sourceIndex])
    {
        if (s[sourceIndex] != del)
        {
            s[sinkIndex] = s[sourceIndex];
            sinkIndex++;
        }
        else
        {
            printf("[%d,%d] %s\n", sourceIndex, sinkIndex, s);
        }
        sourceIndex++;
    }
    s[sinkIndex]='\0';
    printf("result: %s\n",s);
}
```

10. (5 points) What is the output of this program?

```
#include <stdio.h>

void main(void)
{
    char data[] = "Batman";
    data[0] = 'c';
    char *linePt = &data[3];
    *linePt = 'v';
    printf("[%s], [%s]\n", data, linePt);
}
```

11. (6 points) What is the output of this program?

```
#include <stdio.h>

struct Point
{
    int x;
    int y;
};

struct Point incPoints(struct Point p1, struct Point *p2)
{
    p1.x++;
    p1.y++;
    p2->x++;
    p2->y++;
    return p1;
}

void main(void)
{
    struct Point a = {2, 4};
    struct Point b = {5, 1};
    struct Point c = incPoints(a, &b);
    printf("a=(%d, %d), b=(%d,%d), c=(%d,%d)\n",
           a.x, a.y, b.x, b.y, c.x, c.y);
}
```

12. (10 points (bonus)) What is the output of this program?

```
#include <stdio.h>
#include <string.h>

char *findSubstring(char *str, char *target)
{
    int len = strlen(target);
    int n = 0;
    while (*str)
    {
        printf("%c%c ",*str, *(target+n));
        if ( *(target+n) == *str)
        {
            n++;
            if (n == len) return (str-len)+1;
        }
        else
        {
            str -= n;
            n = 0;
        }
        str++;
    }
    return NULL;
}

void main(void)
{
    findSubstring("BANDANA", "ANA");
}
```



13. (10 points (bonus)) What is the output of this program?

```

#include <stdio.h>

void swap(char a[], int i, int j)
{
    char tmp = a[i];
    a[i] = a[j];
    a[j] = tmp;
}

void siftDown(char a[], int i, int n)
{
    int left = 2*i+1;
    int right = 2*i+2;
    int largest = i;

    if(left < n && a[left] > a[largest])
    {
        largest = left;
    }
    if(right < n && a[right] > a[largest])
    {
        largest = right;
    }

    if(largest != i)
    {
        swap(a, i, largest);
        siftDown(a, largest, n);
    }
}

void heapsort(char a[], int n)
{
    int i;
    for(i = (n-2)/2; i >= 0; i--)
    {
        printf("Heaping: %s, i=%d\n", a, i);
        siftDown(a, i, n);
    }
    printf(" Heaped: %s\n", a);
    for(i = n-1; i > 0; i--)
    {
        printf("Sorting: %s, i=%d\n", a, i);
        swap(a, 0, i);
        siftDown(a, 0, i);
    }
}

void main(void)
{
    char data[] = "LAMTOS";
    heapsort(data, 6);
    printf(" Sorted: %s\n", data);
}

```