CS 246
C and Unix
Exam 2

Name	

Question	Points	Score
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	5	
8	5	
9	20	
10	10	
Total:	100	

1. (10 points) Write a complete C program that reads characters from standard input (until end of file) and writes them to standard output, omitting all vowels (a, e, i, o, and u).

Solution:

```
#include <stdio.h>

int main() {
  int c;

  while ((c = getchar()) != EOF) {
    switch (c) {
    case 'a': case 'e': case 'i': case 'o': case 'u':
    case 'A': case 'E': case 'I': case 'O': case 'U':
      break;
    default:
      putchar(c);
    }
  }
}
```

2. (10 points) Write a C function

   `void print_chars(int n, char c);`

   that prints the character c to standard output n times.

Solution:

```
void print_chars(int n, char c) {
  for (int i = 0; i < n; i++) {
    putchar(c);
  }
}
```

3. (10 points) Write a C function

```
void triangle(int n, char c);
```

that uses the function from the previous problem to print a hollow triangle as shown below. This function should have exactly one loop. Hint: printchars can print spaces as well as other characters.

`triangle(5, '*')` will print

```
*****
*   *
* *
**
*
```

and `triangle(2, '$')` will print

```
$$
$
```

Solution:

```
void triangle(int n, char c) {
  print_chars(n, c);
  putchar('\n');
  for (int i = n - 1; i > 1; i--) {
    putchar(c);
    print_chars(i - 2, ' ');
    putchar(c);
  putchar('\n');
  }
  putchar(c);
  putchar('\n');
}
```

4. (10 points) Write a C function

```
double sum(int n, double a[n]);
```

that computes the sum of the items in the array.

Solution:

```
double sum(int n, double a[n]) {
  double s = 0;
  for (int i = 0; i < n; i++) {
    s += a[i];
  }
  return s;
}
```

5. (10 points) Write a C function

```
double read_numbers(int n, double a[n]);
```

that reads n numbers from standard input into the array.

Solution:

```
void read_numbers(int n, double a[n]) {
  for (int i = 0; i < n; i++) {
    scanf("%lf", &(a[i]));
  }
}
```

6. (10 points) Write a complete C program that uses the functions from the previous two problems to read numbers from standard input and print their sum. The number of numbers to read will be given to the program as a command line argument. Do not repeat the code for the functions.

Solution:

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

-- Definitions of sum and read_numbers go here --

  int main(int argc, char *argv[]) {
  if (argc != 2) {
    fprintf(stderr, "usage: sum n\n");
    exit(1);
  }

  int n = atoi(argv[1]);
  double a[n];
  read_numbers(n, a);
  printf("%f\n", sum(n, a));
}
```

7. (5 points) Convert 37 from decimal (base 10) to binary (base 2).

Solution: 100101

8. (5 points) Convert 110101 from binary to decimal.

Solution: 53

9. (20 points) Give the result of each operation (the numbers are in binary).
   (a) `11101111 & 00110111`

   Solution: 00100111

   (b) `11101111 | 00110111`

Solution: 11111111

(c) `11101111 ^ 00110111`

Solution: 11011000

(d) `~00110111`

Solution: 11001000

10. (10 points) Give the result of each operation (the numbers are in decimal).

(a) `8 << 3`

Solution: 64

(b) `15 >> 2`

Solution: 3