1 MATLAB

1.1 User I/O and data types

```
prompt = "Enter a value: ";
x = input(prompt);
txt = input("Enter some text: ", "s");

formatSpec = '%s, %.3f doubled is %.3d';
fprintf(formatSpec, txt, x, 2*double(x))
```

1.2 Loops

```
for index = initVal:step:endVal
   statements
   break
end
while a == b || ~(b > c && c <= d)
    statements
end
```

1.3 Matrices and vectors

```
mtr = zeros(2,4);
vct = [1,2,3,4,5,6,7,8];

mtr(1,1) = vct(8);
```

1.4 Conditional execution

```
if expression
    statements
elseif expression
    statements
else
    statements
end
```

```
switch switch_expression
   case case_expression
      statements
   case case_expression
      statements
    ...
   otherwise
      statements
end
```

1.5 Custom functions

```
function [a,b] = ftn(a,b)
    statements
end

[a,b] = ftn(a,b)
```

1.6 Other functions

```
num = abs(num);
num = mod(num,2);
tf = strcmp(s1, s2)

for i = 1:size(mtr,1)
    for j = 1:size(mtr,2)
        statements
    end
end
```

2 C

2.1 User I/O

```c
int main(void) {
  int n;
  printf("Enter a number: "); // requires stdio.h
  scanf("%d", &n); // requires stdio.h
  printf("The number is %d.\n", n);
  return 0;
}
```

2.2 Loops

```c
  for (int i = 0; i < 10; i++) {
    printf("%d\n", i);
  }

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

2.3 Matrices

```c
  int matrix[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

  int *matrix2 = malloc(sizeof(int) * 2 * 2); // requires stdlib.h
  matrix2[0] = 1;
  matrix2[1] = 2;
  matrix2[2] = 3;
  matrix2[3] = 4;
  free(matrix2);
```

2.4 Conditional execution

```c
  int a = 1;
  if (a == 1) {
    printf("a is 1\n");
  } else if (a == 2) {
    printf("a is 2\n");
  } else {
    printf("a is neither 1 nor 2\n");
  }
```

```c
  int b = 1;
  switch (b) {
    case 1:
      printf("b is 1\n");
      break;
    case 2:
      printf("b is 2\n");
      break;
    default:
      printf("b is neither 1 nor 2\n");
      break;
  }
```

2.5 Custom function and rng

```c
int rand_num(int min, int max) {
  srand(time(NULL)); // requires time.h
  return min + (rand() % (max - min + 1)); // requires stdlib.h
}
```

2.6 String functions

```c
// requires string.h
  strcat(dest, src); // concatenate src to end of dest
  strcpy(dest, src); // copy src to dest, overwriting
  int len = strlen(str); // returns length of string
```