Foundation Exam Reference Sheet

stdlib.h functions

```
// Allocates size bytes and returns a pointer to the beginning
// of the block of memory allocated.
void* malloc (size_t size);

// Allocates an array of nitems, each which is size bytes
// big, sets all bits to 0 and returns a pointer to the
// beginning of the block of memory allocated.
void* calloc(size_t nitems, size_t size);

// Attempts to resize the memory block pointed to by ptr to be
// size bytes and returns a pointer to the beginning of the
// block of memory allocated.
void* realloc(void* ptr, size_t size);

// Deallocates memory pointed to by ptr.
free(void* ptr);

// Seeds the random number generator used by the rand function
// with seed.
void srand(unsigned int seed);

// Returns a pseudo-random number in the range of 0 to
// RAND_MAX (usually 32767).
int rand(void);

// Returns the absolute value of x.
int abs(int x);

// Returns the string pointed to by str to its equivalent
// integer value, so long as the string pointed to by str is
// a valid integer.
int atoi(const char* str);

// Returns the string pointed to by str to its equivalent
// float value, so long as the string pointed to by str is
// a valid floating point number.
int atof(const char* str);
```

math.h functions

```
// Returns x raised to the power y.
double pow(double x, double y);


// Returns the square root of x as long as x >= 0.
double sqrt(double x);

// Returns the absolute value of x.
double fabs(double x);

// Returns e raised to the power x.
double exp(double x);

// Returns the natural log of x, so long as x > 0.
double log(double x);
```

string.h functions

```
// Returns a negative integer if the string pointed to by str1
// comes before the string pointed to by str2 lexicographically,
// 0 if both strings are equal and a positive integer if the
// string pointed to by str1 comes after the string pointed to
// by str2 lexicographically.
int strcmp(const char* str1, const char* str2);

// Returns the length of the string pointed to by str.
int strlen(const char* str);

// Copies the contents of the string pointed to by src into
// the string pointed to by dest and returns a pointer to the
// memory address where the string was copied.
char* strcpy(char* dest, const char* src);

// Appends the contents of the string pointed to by src
// to the string pointed to by dest and returns a pointer to
// the memory address of the beginning of the concatenated
// string.
char* strcat(char* dest, const char* src);
```

Summation Formulas

$\sum_{i=1}^{n} c = cn \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$, $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$, $\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$, $\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}, |x| < 1$