For each node:

- Insert that node into "newhead"

$q \rightarrow \text{head} = \text{newhead}


```plaintext
newhead \overset{?}{=} \quad \text{loop until}
\quad \text{Save} = \text{head} \rightarrow \text{next}
\quad \text{head} \rightarrow \text{next} = \text{newhead}
\quad \text{newhead} = \text{head}
\quad \text{head} = \text{Save}
```
char buffer[10];
char *p = buffer;
buffer[0] = 'A';
buffer[1] = 'B';
P[0] = 'C';
P[1] = 'D';
char str[20];
strcpy — copies chars until it hits a \0 or crashes

p = malloc(strlen(s) + 1)
if (p != NULL) {
    strcpy(p, s);
}

OK because you allocated enough space.
```c
char buffer[6] = { 'c', 'a', 'l' );
strncpy(buffer, "mooose", 5 );
```

`buffer[5] = \0;` "cal" ok

"mooese" ok

"lizard"

```c
S 1 1 1 2 2 2 3
```

buffer 1 1 1 2 2 2 3 `uh-oh`
stlcpy
Characters

0-9 ( ) , . ; + -

A-Z a-z

ω Ω
ANSI — American National Standards
NIST — National Inst. of Standards
IETF
ISO
IEEE
EBCDIC — punch cards
ASCII — American Stand. Code for Inf. Interchange

Unicode Consortium

character ↔ integer

"Codepoint"
I want to store a character in a file (or transmit it across a network or...)

Systems for storing ints "encodings"
Encoding Attempt #1

16-bit integers

$2^{16}$ possible characters

$U+0041$  
'A'

\[ \begin{array}{c|c}
00000000 & 0100 0000 \\
\text{first this byte} & \text{then this byte}
\end{array} \]

roughly UCS-2 BE
Attempts

| r bits | 8 bits |

have a marker that says "we need more bytes"

Roughly UTF-16
English in UCS-2

Dear mom

0x44
0x65
0x61

0x00
0x44
0x00
0x65
Ken Thompson (Unix)

UTF-8
- Throw away fixed memory size for a char
- Get rid of NULL bytes
- Get rid of endian issues