

IMPORTANT: Do not open this packet until the exam starts. After the exam starts, you can tear apart the pages in this packet if you want. This packet will not be graded, but you must turn all three pages in at the end of the exam. You may use any space in this packet as scratch paper.

C Function Definitions

```
int atoi(const char *nptr);
```

atoi() reads the (null-terminated) string at nptr and converts it to an integer.

Example: atoi("-5") == -5

Example: atoi("12") == 12

```
char *fgets(char *s, int size, FILE *stream);
```

fgets() reads in at most one less than size characters from stream and stores them into the buffer pointed to by s. Reading stops after an EOF or a newline. If a newline is read, it is stored into the buffer. A terminating null byte ('\0') is stored after the last character in the buffer.

```
char *gets(char *s);
```

gets() reads a line from stdin into the buffer pointed to by s until either a terminating newline or EOF, which it replaces with a null byte ('\0').

```
int printf(const char *format, ...);
```

printf() produces output according to the format string format.

Q7 Returnless**(0 points)**

The C code is reproduced here for your convenience. This page will not be graded.

```
1 int main(void) {  
2     int i;  
3     char input[16];  
4     int stored_nums[4];  
5  
6     for (i = 0; i < 4; i++) {  
7         printf("Enter stored number for %d: ", i);  
8         gets(input);  
9         stored_nums[i] = atoi(input);  
10    }  
11  
12    while (i >= 0) {  
13        printf("Which stored number do you want to print? ");  
14        gets(input);  
15        i = atoi(input);  
16        if (i >= 0) {  
17            printf("%d\n", stored_nums[i]);  
18        }  
19    }  
20    return 0;  
21 }
```

Q8 161 Meets 61A**(0 points)**

The C code is reproduced here for your convenience. This page will not be graded.

```
1 void add_letter(int i, char *buf) {
2     char word[4];
3     printf("Enter Word %d:\n", i);
4     fgets(word, 4, stdin);
5     buf[i] = word[0];
6     if (i > 0) {
7         add_letter(i - 1, buf);
8     }
9 }
10
11 void make_acronym(void) {
12     char result[4];
13     add_letter(4, result);
14     printf("%s\n", result);
15 }
16
17 void word_games(void) {
18     make_acronym();
19 }
20
21 int main(void) {
22     word_games();
23     return 0;
24 }
```