

16.216: ECE Application Programming

Practice Problems: 1-D Arrays, Pointer Arithmetic, and Strings

1-D Arrays

1. What does each of the following programs print?

- a.

```
int main() {
    int arr[15];
    int i;
    for (i = 0; i < 15; i++)
        arr[i] = i * 3;
    for (i = 15; i > 0; i--)
        printf("%d\n", arr[i-1]);
    return 0;
}
```
- b.

```
int main() {
    double vals[] = {1.2, 3.5, 4, -7.8, 6.7, 8.7};
    int i;
    int n = sizeof(vals) / sizeof(double);

    printf("n = %d\n", n);
    printf("vals[1] + vals[3] = %lf\n", vals[1] + vals[3]);
    printf("vals[2] - vals[5] = %lf\n", vals[2] - vals[5]);
    printf("vals[n-3] + vals[0] = %lf\n", vals[n-3] + vals[0]);
    return 0;
}
```
- c.

```
int main() {
    int a[8] = {1, 2, 7, 0, 4, 5, 3, 6};
    int b[8] = {16, 216, 201, 202, 2011};
    int i;
    for (i = 0; i < 8; i++){
        printf("%d %d\n", a[i], b[a[i]]);
    }
    return 0;
}
```

1 (cont.) What does each of the following programs print?

```
d. void f(int arr[], int n) {
    int i;
    for (i = 0; i < n; i++)
        arr[i] = arr[n-i-1];
}

void printArray(int arr[], int n) {
    int i;
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int i;
    int q[6] = {1, 1, 2, 3, 5, 8};

    printArray(q, 6);

    f(q, 6);
    printArray(q, 6);

    f(q, 4);
    printArray(q, 6);

    return 0;
}

e. int f(int x[]) {
    int t = 0;
    int i = 0;
    while (x[i] != -1) {
        t += x[i];
        i++;
    }
    return t / i;
}

int main() {
    int a1[] = {1, 2, 3, 4, 5, -1};
    int a2[] = {-2, -4, -5, -1, 3, 2, 1};
    int a3[] = {10, -1, 20, -1, 30, -1, 40, -1};
    printf("%d %d %d\n", f(a1), f(a2), f(a3));
    return 0;
}
```

2. Write a function to do each of the following tasks:
- `checkIfSorted()`: Given an array of integer values, `a[]`, and the size of the array, `n`, check if the array is sorted from smallest to largest value. If so, return 1; if not, return 0.
 - `countDiv()`: Given an array of integer values, `a[]`, the size of the array, `n`, and a value `v`, count and return the number of values in `a[]` that are divisible by `v`.
 - `fillArray()`: Given an empty array of double-precision values, `d[]`, and the total size of the array, `n`, repeatedly read values from the standard input and store them in `d[]` until one of two conditions occurs:
 - The user enters the value 0 (which should be stored in the array).
 - The array is completely full.
 Once done, your function should return the number of values actually stored in the array.

Pointer arithmetic

3. What does each of the following programs print?

```
a. int main() {
    int i;
    int arr[10];
    int *p = arr;
    for (i = 0; i < 10; i++) {
        *p = i * i;
        p++;
    }
    for (i = 0; i < 10; i++)
        printf("%d\n", arr[i]);
    return 0;
}
```

```
b. int main() {
    double *d;
    double p[]={49.1, 90.4, 76.6, 85.3, 78.4, 80.2, 70.0};
    d = p + 2;

    printf("%lf\n", *d);
    d--;
    printf("%lf\n", *d);
    d += 4;
    printf("%lf\n", *d);
    d - 2;
    printf("%lf\n", *d);
    return 0;
}
```

3 (cont.) What does the following program print?

```
c. int main() {
    int i;
    char str[] = "ece application programming";
    char *sp = str;
    for (i = 0; i < 14; i++) {
        (*sp) -= 32;    // Convert character that sp points to
                       // to uppercase
        sp += 2;
    }
    printf(str);
    return 0;
}
```

4. Write code to implement your own version of each of the following string functions, using pointers to deal with each string:

- int strlen(char *s);
- char *strcpy(char *dest, char *source);
- int strncmp(char *s1, char *s2, int n);

Strings

5. What does each of the following programs print?

```
a. int main() {
    int i;
    char str[] = "1234567890abcdefghij";
    for (i = 1; i < strlen(str); i *= 2) {
        printf("%c\n", str[i]);
    }
    return 0;
}
```

```
b. int main() {
    char s1[] = "String1";
    char s2[] = "String2";
    int i;
    for (i = 1; i < strlen(s1); i++) {
        if (strncmp(s1, s2, i) == 0)
            printf("Match\n");
        else {
            printf("No match\n");
            break;
        }
    }
    return 0;
}
```

5 (cont.) What does the following program print?

```
c. int main() {
    char s1[20] = "";
    char s2[20] = "";
    strcat(s1, "ab");
    strcat(s2, "ac");
    strcat(s1, s2);
    strcat(s2, s1);
    strncat(s1, s2, 3);
    strncat(s2, s1, 3);
    printf("%s %s\n", s1, s2);
    return 0;
}
```

6. Write a function to do each of the following tasks:

- a. `buildString()`: Given a character array, `str[]`, and the length of the array, `n`, repeatedly read strings from the standard input and store them in `str[]`, ensuring there is a single space between each string, until either the user enters Ctrl-Z (end of file) or `str[]` does not have enough remaining room to hold the next string. For example, if `n = 10`:
- User enters “one word” → `str` holds “one word”
 - User enters “three words” → `str` holds “three” (not enough space to hold both words)

The function should return the actual length of the string stored in `str[]`

- b. `longestMatch()`: Given two strings, `s1` and `s2`, return the length of the longest matching character sequence between the two, starting with the first character of each. For example:
- `s1 = “string”, s2 = “other”` → function returns 0
 - `s1 = “string”, s2 = “stuff”` → function returns 2
 - `s1 = “string”, s2 = “strings”` → function returns 6
- c. `copyFromPosn()`: Given two strings, `dest` and `source`, as well as an integer `pos`, copy all characters from `source` into `dest`, starting at position `pos` and ending with the end of the source string. Assume `pos` is a valid position within `source`, and there is enough room in `dest` to hold the source string—you do not need to check for errors. For example:
- `source = “string”, pos = 0` → `dest = “string”`
 - `source = “string”, pos = 3` → `dest = “ing”`