

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



BÁO CÁO THỰC TẬP

Sinh viên :	Nguyễn Anh Tuấn
MSSV :	20021594
Lớp :	QH 2020 –ĐACLIC2
Đơn vị thực tập :	Công ty FPT Software

Hà Nội, 8 -2023

LỜI NÓI ĐẦU

Kính gửi các thầy cô trong khoa Viện Điện tử - Viễn thông!

Lời đầu tiên em xin được cảm ơn tới tất cả các thầy cô trong Viện Điện tử - Viễn thông, các thầy cô trong Ban lãnh đạo viện, đã tạo cơ hội cho chúng em có đợt thực tập thực tế tại FPT Software. Em xin chân thành cảm ơn thầy Hoàng Gia Hưng, cùng các thầy cô trong viện đã tận tình hướng dẫn chúng em trong đợt thực tập này.

Đồng thời chúng em xin chân thành cảm ơn Ban lãnh đạo Công ty FPT Software và các anh chị đã trực tiếp hướng dẫn chúng em hoàn thành tốt quá trình thực tập này.

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU CHỨC NĂNG, NHIỆM VỤ, CƠ CẤU TỔ CHỨC CỦA ĐƠN VỊ TIẾP NHẬN

1.1. Tập đoàn FPT.....	3
1.2. Lịch sử hình thành phát triển.....	3
1.3. Cơ cấu tổ chức.....	5
1.4. Công ty FPT Software.....	5
1.5. Lịch sử phát triển.....	6
1.6. Cơ cấu tổ chức.....	6
1.7. Tình hình hoạt động.....	7

CHƯƠNG 2: NỘI DUNG THỰC TẬP.....8

2.1. Khóa học Writing Email.....	8
2.2. Khóa học Campuslink C.....	8
2.3. Khóa học MCP.....	23

CHƯƠNG 3: NHẬN XÉT, ĐỀ XUẤT.....29

3.1. Ưu điểm.....	29
3.2. Nhược điểm.....	29

CHƯƠNG 4: KẾT LUẬN.....30

CHƯƠNG 1: GIỚI THIỆU CHỨC NĂNG, NHIỆM VỤ, CƠ CẤU TỔ CHỨC CỦA ĐƠN VỊ

1.1. Tập đoàn FPT

Tập đoàn FPT (tiếng Anh: *FPT Corporation*), tên chính thức là **Công ty Cổ phần FPT**, được biết đến rộng rãi dưới cái tên **FPT**, là công ty thuộc top 20 doanh nghiệp tư nhân lớn nhất tại Việt Nam, với ba lĩnh vực kinh doanh cốt lõi gồm: Công nghệ, Viễn thông và Giáo dục.

1.2. Lịch sử hình thành phát triển

Ngày 13 tháng 9 năm 1988, FPT được thành lập với tên gọi Công ty Công nghệ Thực phẩm hoạt động trong lĩnh vực công nghệ sây, công nghệ thông tin, công nghệ tự động hóa. (Chữ gốc FPT ban đầu có nghĩa là The Food Processing Technology Company - Công ty Công nghệ Thực phẩm).

Ngày 27 tháng 10 năm 1990, được đổi tên thành The Corporation for Financing Promoting Technology - Công ty Đầu tư và Phát triển Công nghệ với hoạt động kinh doanh cốt lõi là CNTT.[2]

Năm 1998, FPT trở thành 1 trong 4 nhà cung cấp dịch vụ Internet đầu tiên tại Việt Nam, tạo bước phát triển đột phá cho lĩnh vực này tại Việt Nam.

Tháng 4 năm 2002, FPT trở thành công ty cổ phần.

Ngày 8 tháng 9 năm 2006, Thủ tướng Chính phủ Việt Nam ký quyết định thành lập trường Đại học FPT trực thuộc Công ty FPT. Hiệu trưởng của trường là Tiến sĩ Lê Trường Tùng, còn Chủ tịch Hội đồng Quản trị là Phó giáo sư, Tiến sĩ Trương Gia Bình.

Ngày 24 tháng 10 năm 2006, FPT đã công bố quyết định phát hành thêm cổ phiếu cho hai nhà đầu tư chiến lược là Quỹ đầu tư Texas Pacific Group (TPG) và Intel Capital. FPT nhận được một khoản đầu tư là 36,5 triệu USD thông qua quỹ đầu tư TPG Ventures và Intel Capital.[5]

Ngày 18 tháng 11 năm 2006, Tập đoàn Microsoft và công ty FPT ký thoả thuận liên minh chiến lược.

Ngày 13 tháng 12 năm 2006, cổ phiếu FPT lên sàn chứng khoán Thành phố Hồ Chí Minh (HoSE).

Ngày 25 tháng 12 năm 2006, Chủ tịch HĐQT Công ty FPT đã ký quyết hợp nhất các Công ty TNHH Hệ thống Thông tin FPT (FIS), Công ty TNHH Giải pháp Phần mềm FPT (FSS) và Trung tâm dịch vụ ERP (FES) kể từ ngày 01/01/2007. Công ty hợp nhất có tên là Công ty TNHH Hệ thống Thông tin FPT.

Ngày 1 tháng 1 năm 2007, FPT thành lập Công ty TNHH Bán lẻ FPT với mô hình Công ty TNHH một thành viên.

Ngày 13 tháng 3 năm 2007, thành lập Công ty Cổ phần Quảng cáo FPT (FPT Promo JSC) và Công ty TNHH Phần mềm Châu Á Thái Bình Dương đặt tại Singapore (FAPAC).

Ngày 19 tháng 12 năm 2008, Công ty FPT công bố được chấp thuận đổi tên từ "Công ty Cổ phần Phát triển và Đầu tư Công nghệ" thành "Công ty Cổ phần FPT" viết tắt là "FPT Corporation".[7]

Ngày 24 tháng 12 năm 2008, Công ty FPT công bố quyết định bổ nhiệm Tổng Giám đốc mới là ông Nguyễn Thành Nam thay ông Trương Gia Bình. Ông Nam là thành viên sáng lập Công ty FPT, Chủ tịch HĐQT của Công ty Phần mềm FPT (FSOFT).

Tháng 2 năm 2011, Hội đồng Quản trị Công ty FPT có nghị quyết bổ nhiệm ông Trương Đình Anh làm Tổng giám đốc thay thế cho ông Nguyễn Thành Nam.[8]

Ngày 31 tháng 7 năm 2013, Hội đồng Quản trị Công ty FPT đã thông qua nghị quyết bổ nhiệm ông Bùi Quang Ngọc, Tiến sĩ về cơ sở dữ liệu, làm Tổng Giám đốc FPT thay thế ông Trương Gia Bình. Ông Ngọc là một trong các thành viên sáng lập và hiện là Phó Chủ tịch Hội đồng Quản trị FPT.

Năm 2014, FPT là công ty CNTT đầu tiên của Việt Nam mua lại 1 công ty CNTT nước ngoài, RWE IT Slovakia (đơn vị thành viên của Tập đoàn năng lượng hàng đầu châu Âu, RWE).[9]

Tháng 8 năm 2017, FPT chuyển nhượng 30% vốn tại Công ty Cổ phần Bán lẻ kỹ thuật số FPT (FPT Retail) cho Vina Capital và Dragon Capital, giảm tỷ lệ sở hữu của Công ty tại FPT Retail xuống còn 55%.[10]

Tháng 9 năm 2017, FPT chuyển nhượng 47% vốn tại Công ty Cổ phần Thương mại FPT (FPT Trading) cho Tập đoàn Synnex (Đài Loan), giảm tỷ lệ sở hữu của Công ty tại FPT Retail xuống còn 48%.[11]

Tháng 7 năm 2018, FPT mua 90% cổ phần của Intellinet Consulting (Intellinet), một trong những công ty tư vấn công nghệ có tốc độ tăng trưởng nhanh nhất tại Mỹ.

Tháng 3 năm 2019, FPT đã bổ nhiệm ông Nguyễn Văn Khoa - Phó Tổng Giám đốc FPT kiêm Tổng Giám đốc Công ty TNHH Hệ thống Thông tin FPT - đảm trách vị trí Tổng Giám đốc FPT, thay thế ông Bùi Quang Ngọc hết nhiệm kỳ.[12]

Tháng 6 năm 2020, FPT ký kết thỏa thuận hợp tác chiến lược với Viện nghiên cứu trí tuệ nhân tạo hàng đầu thế giới Mila, mở ra cơ hội cọ xát, nâng cao năng lực công nghệ cho cộng đồng công nghệ trẻ của Việt Nam.[13]

Tháng 5 năm 2021, FPT đầu tư chiến lược vào nền tảng quản trị doanh nghiệp số 1 Việt Nam - Base.vn thúc đẩy chuyển đổi số doanh nghiệp SMEs.[14]

Tháng 7 năm 2021, FPT mua công ty Intertec International - doanh nghiệp cung cấp dịch vụ công nghệ thông tin có hơn 20 năm kinh nghiệm ở châu Mỹ Latinh.[15]

Tháng 8 năm 2021, FPT khởi xướng chương trình FPT eCovax - Vaccine số cho doanh nghiệp, hỗ trợ hơn 3.000 doanh nghiệp vận hành, kinh doanh không gián đoạn và tăng trưởng bứt phá trong bình thường xanh.[16]

Tháng 9 năm 2021, ông Trương Gia Bình, Chủ tịch Hội đồng Quản trị Tập đoàn FPT đã khởi xướng ý tưởng xây dựng một ngôi trường dành cho các em nhỏ mất cha, mẹ do dịch COVID-19. Tròn một năm sau, Trường Hy Vọng (tại quận Ngũ Hành Sơn, TP. Đà Nẵng) đón nhận 200 học sinh đầu tiên trong năm học 2022-2023.[17]

Tháng 5 năm 2022, FPT tổ chức khởi công xây dựng Tổ hợp giáo dục FPT UniSchool Hà Nam - cơ sở tập trung đào tạo các cấp học từ phổ thông đến cao đẳng với quy mô đào tạo 10.000 người theo học, nâng tổng số cơ sở đào tạo tại 14 tỉnh, thành phố.

Bằng việc mở thêm văn phòng thứ hai tại New York vào tháng 5 năm 2022[18] và khai trương văn phòng đại diện đầu tiên ở Bắc Âu tại Copenhagen, Đan Mạch vào tháng 9 năm 2022[19], FPT đã tăng cường sự hiện diện tại 27 quốc gia trên thế giới.

Tháng 9 năm 2022, Phó Thủ tướng Chính phủ Singapore Heng Swee Keat thăm FPT Software (Công ty thành viên của Tập đoàn FPT), đẩy mạnh hoạt động liên kết, hợp tác giữa các doanh nghiệp khu vực tư nhân giữa Việt Nam và Singapore.[20]

Tháng 10 năm 2022, FPT trở Thành Cổ Đông Chiến Lược Của LTS, Inc. - công ty trong Top 20 công ty tư vấn, quản trị kinh doanh và chuyển đổi số tại Nhật với hơn 20 năm kinh nghiệm.[21]

1.3. Cơ cấu tổ chức

FPT có hệ thống văn phòng tại 27 quốc gia trên thế giới, và hạ tầng viễn thông phủ khắp 59/63 tỉnh, thành phố tại Việt Nam. Công ty đặt trụ sở chính tại Tòa nhà FPT, số 10 Phố Phạm Văn Bạch, Phường Dịch Vọng, Quận Cầu Giấy, Thành phố Hà Nội, với 8 công ty thành viên và 3 công ty liên kết.

8 Công ty thành viên:

- Công ty TNHH Phần mềm FPT (FPT Software)
- Công ty TNHH Hệ thống Thông tin FPT (FPT Information System)
- Công ty Cổ phần Viễn thông FPT (FPT Telecom)
- Công ty Cổ phần Dịch vụ Trực tuyến FPT (FPT Online)
- Công ty TNHH Giáo dục FPT (FPT Education)
- Công ty TNHH Đầu tư FPT (FPT Investment)
- Công ty TNHH FPT Smart Cloud (FPT Smart Cloud)
- Công ty TNHH FPT Digital (FPT Digital)

2 Công ty liên kết:

- Công ty Cổ phần Synnex FPT (Synnex FPT)
- Công ty Cổ phần Bán lẻ Kỹ thuật số FPT (FPT Retail)

1.4. Công ty FPT Software

Công ty TNHH Phần mềm FPT (tiếng Việt: *Công ty TNHH Phần mềm FPT*, nghĩa là 'FPT Software LLC'), hay đơn giản là **FPT Software**, là nhà cung cấp dịch vụ CNTT toàn cầu có trụ sở chính tại Hà Nội, Việt Nam, là công ty con cốt lõi của Tập đoàn FPT. Đây được đánh giá là một trong những công ty dịch vụ phần mềm lớn nhất Việt Nam, với doanh thu 1,87 tỷ USD và hơn 42.000 nhân viên.

1.5. Lịch sử phát triển

FPT Software được thành lập vào năm 1999 bởi 13 thành viên của Tập đoàn FPT, dẫn đầu là Nguyễn Thành Nam (sau này là CEO, Chủ tịch FPT Software rồi CEO Tập đoàn FPT). Năm 2000, nó đã mở hai chi nhánh nước ngoài đầu tiên tại Thung lũng Silicon, Mỹ và Bangalore, Ấn Độ; Cả hai đều đóng cửa sau một năm do thiếu khách hàng. Ban lãnh đạo FPT quyết định chuyển hướng sang thị trường Nhật Bản, điều này thúc đẩy sự tăng trưởng của công ty. Năm 2005, FPT mở chi nhánh đầu tiên tại Nhật Bản và Singapore, sau đó mở rộng sang Paris, Pháp và mở lại chi nhánh tại Mỹ vào năm 2008. Công ty cũng mở trung tâm giao hàng nội địa tại thành phố Hồ Chí Minh (2004) và tại Đà Nẵng (2005).

Năm 2009, FPT Software được tổ chức lại thành công ty cổ phần, với Chủ tịch HĐQT mới được bổ nhiệm là Nguyễn Thành Nam (Nguyên CEO FPT Software) và CEO Bùi Thị Hồng Liên (nguyên CEO FPT Ấn Độ và cựu CEO FPT Nhật Bản).

Năm 2012, FPT Software có sự thay đổi tổ chức lớn đầu tiên, với những thay đổi trong ban giám đốc, mô hình kinh doanh (từ nhiều công ty con sang trung tâm giao hàng và chi nhánh nước ngoài) và chiến lược (từ dịch vụ thuê ngoài công nghệ thông tin truyền thống (ITO) sang dịch vụ định hướng nền tảng thứ ba (SMAC). Tính đến cuối năm 2013, FPT Software cán mốc 100 triệu USD doanh thu và 5.000 nhân sự.

Tháng 2014/400, FPT Software mua lại RWE IT Slovakia, một đơn vị kinh doanh CNTT của RWE, trở thành công ty CNTT Việt Nam đầu tiên tiến hành mua bán và sáp nhập ở nước ngoài. Thỏa thuận này đã giúp FPT Software tiếp cận với lĩnh vực quản lý năng lượng và bí quyết về công nghệ SAP với hơn 2016 chuyên gia địa phương. Năm 2016, FPT Software đạt doanh thu 10 triệu USD và 000.20 nhân viên, đưa công ty sánh ngang với 126 công ty dịch vụ CNTT hàng đầu Ấn Độ. FPT Nhật Bản cũng đạt doanh thu 100 triệu USD, trở thành Chi nhánh nước ngoài đầu tiên của FPT Software vượt mốc doanh thu trên < > triệu USD.

Năm 2017, được tài trợ bởi Chủ tịch Tập đoàn FPT Trương Gia Bình, công ty đã khởi động các chiến dịch lớn về chuyển đổi kỹ thuật số và chiến lược săn cá voi, giúp công ty thiết lập quan hệ với hơn 40 tập đoàn lớn trên toàn cầu, trong đó hơn 20 công ty nằm trong danh sách Fortune Global 500 như Airbus, Siemens, UPS, v.v. Tính đến cuối năm 2017, FPT Software đã có 75 đối tác trong danh sách Fortune Global 500. Tháng 2018/90, FPT Software mua lại 150% cổ phần Intellinet, một công ty tư vấn có trụ sở tại Mỹ với 30 chuyên gia tư vấn cấp cao với doanh thu 45 triệu USD. Thương vụ ước tính đạt 50-10 triệu USD khi FPT Software mua lại 1% còn lại trong ba năm tới. FPT Japan đạt quy mô 000.40 nhân sự (không kể ở nước ngoài), trở thành công ty Việt Nam lớn nhất Nhật Bản và nằm trong Top < > công ty CNTT Nhật Bản.

Cơ cấu tổ chức

FPT Software có mặt tại 26 quốc gia & vùng lãnh thổ với 59 văn phòng.

Vùng	Nước	Tên chi nhánh	Văn phòng
Châu Á-Thái Bình Dương	Việt Nam	FPT Software Việt Nam	Hà Nội, Thành phố Hồ Chí Minh, Đà Nẵng, Quy Nhơn, Cần Thơ
	Các nước châu Á - Thái Bình Dương	FPT Châu Á – Thái Bình Dương	Singapore Malaysia: Kuala Lumpur Úc: Sydney, Melbourne Philippines: Cebu, Manila Indonesia: Jakarta Việt Nam: Hà Nội, Hồ Chí Minh, Đà Nẵng, Quy Nhơn, Cần Thơ Trung Quốc: Thượng Hải, Hồng Kông Thái Lan: Bangkok Myanmar: Yangon
Châu Á	Nhật Bản	FPT Nhật Bản	Tokyo, Sapporo, Shizuoka, Yokohama, Nagoya, Fukuoka, Osaka, Okinawa, Toyota, Hiroshima
	Hàn Quốc	FPT Hàn Quốc	Seoul
	Đài Loan		Đài Bắc
Châu Mỹ	Hoa Kỳ	FPT Mỹ	Hoa Kỳ: Chicago, Los Angeles, San Francisco, Dallas, Denver, Seattle, Minneapolis, Atlanta, Detroit, Thành phố New York Canada: Ontario
	Canada		Ontario
	Columbia		Columbia
Châu Âu	Đức	FPT Châu Âu	Đức: Essen Pháp: Paris, Toulouse Slovakia: Kosice Bỉ: Mechelen Vương quốc Anh: London Cộng hòa Séc: Strašnice Đan Mạch: Kongens Lyngby
Trung Đông	Các Tiểu Vương Quốc Ả Rập Thống Nhất	FPT UAE	Dubai

Các công ty con

Tên công ty	Mối quan hệ với FPT Software	Ngành nghề kinh doanh
Công ty Cổ phần Kỹ thuật Công nghệ Trandata (TDT)	Chi nhánh	Công nghệ, Phần mềm máy tính, Bán lẻ
Công ty TNHH Dịch vụ Xử lý số FPT	Chi nhánh	Phụ trách BPO ^[1]
Intellinet Tư vấn LLC.	Công ty con cấp hai	Tư vấn công nghệ
UTOP Ltd.	Chi nhánh	Chương trình khách hàng thân thiết dựa trên Blockchain

Tình hình hoạt động

Không bao gồm bộ phận văn phòng, các đơn vị phân phối của FPT Software có thể được chia thành ba bộ phận: chín bộ phận bán hàng để giám sát thị trường nước ngoài, 14 bộ phận kinh doanh trong các ngành và sản phẩm dọc cụ thể và mười bộ phận phát triển phụ trách phát triển kinh doanh trong các dịch vụ cụ thể.

CHƯƠNG 2: NỘI DUNG THỰC TẬP

2.1. Khóa học Writing Email

Dạy về cách viết Email và Cv chuyên nghiệp

2.2. Khóa học Campuslink C

Assignment 1 - Variable

Viết chương trình tìm số fibonacci thứ N ($1 \leq N \leq 100$).

Phương pháp giải :

```
int a1 = 1, a2 = 1;
    if (n == 1 || n == 2)
        return 1;
    int i = 3, a;
    while (i <= n)
    {
        a = a1 + a2;
        a1 = a2;
        a2 = a;
        i++;
    }
    return a;
```

Assignment 2 - Function

Viết một chương trình quản lý mảng số nguyên với các yêu cầu như sau:

Khi chạy chương trình sẽ hiện lời giới thiệu chương trình “CT Quản Lý dãy số bằng mảng” và hướng dẫn sử dụng với các tính năng của chương trình:

Mảng đó là 100 phần tử -> biến global (type là uint8_t -> FF = 255)

Giá trị đúng là trong khoảng 0 -100:

- Nhập ‘c’ để clear data của mảng cũ và cho người dùng nhập.
- Nhập ‘p’ để in dãy (Nếu dãy chưa có giá trị thì báo dãy trống).
- Nhập ‘i’ để thêm 1 phần tử vào dãy với vị trí k (nếu có rồi thì không cho nữa).

- Nhập 'd' để xóa 1 phần tử tại vị trí thứ k (Nếu vị trí k không chứa giá trị thì báo ko có phần tử k).
- Nhập 's' để in ra dãy sắp xếp dãy số theo chiều tăng dần.
- Nhập 'x' để in ra dãy sắp xếp theo chiều giảm.
- Nhập 't' để tìm kiếm 1 số x có trong dãy không (x nhập từ bàn phím)? -> nếu có thì đưa ra vị trí.
- Nhập 'a' để in ra màn hình số lớn nhất.
- Nhập 'w' để in ra màn hình số nhỏ nhất.
- Nếu nhập 'e' thì thoát chương trình.

Phương pháp giải:

```
static int32_t arr[ARRAY_SIZE];
static int32_t count = 0;

void inputArray() {
    count = 0;
    printf("Enter the number of elements in the array:");
    scanf("%d", &count);

    if (count > ARRAY_SIZE) {
        printf("The number of elements exceeds the array size");
        count = ARRAY_SIZE;
    }

    printf("Enter the elements of the array:\n");
    int32_t i;
    for (i = 0; i < count; i++) {
        printf("arr[%d] = ", i);
        scanf("%d", &arr[i]);
    }

    printf("Array import successful\n");
}
```

```

void clearArray() {
    int32_t i;
    for (i = 0; i < ARRAY_SIZE; i++) {
        arr[i] = 0;
    }
    count = 0;
    printf("Deleted\n");
    printf("Enter the number of elements in the array:\n");
    scanf("%d", &count);

    if (count > ARRAY_SIZE) {
        printf("The number of elements exceeds the array size\n");
        count = ARRAY_SIZE;
    }

    printf("Enter the elements of the array:\n");
    for (i = 0; i < count; i++) {
        printf("arr[%d] = ", i);
        scanf("%d", &arr[i]);
    }

    printf("Array import successful\n");
}

void printArray() {
    if (count == 0) {
        printf("Empty Array\n");
        return;
    }

    printf("Elements in array:");
    int32_t i;

```

```

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

void insertElement() {
    if (count == ARRAY_SIZE) {
        printf("Array full, do not import more element\n");
        return;
    }

    int32_t value, position;
    printf("enter the element you want to add:");
    scanf("%d", &value);
    printf("Enter position you want to insert:");
    scanf("%d", &position);

    if (position < 0 || position > count) {
        printf("inappropriate position\n");
        return;
    }

    int32_t i;
    for (i = count; i > position; i--) {
        arr[i] = arr[i - 1];
    }

    arr[position] = value;
    count++;
}

```

```

    printf("added element %d to position %d\n", value, position);
}

void deleteElement() {
    if (count == 0) {
        printf("Empty array . No element to delete\n");
        return;
    }

    int32_t position;
    printf("Enter the possition you want to delete");
    scanf("%d", &position);

    if (position < 0 || position >= count) {
        printf("inappropriate position\n");
        return;
    }

    int32_t deletedValue = arr[position];

    int32_t i;
    for (i = position; i < count - 1; i++) {
        arr[i] = arr[i + 1];
    }

    count--;

    printf("Deleted element %d at position %d.\n", deletedValue, position);
}

void sortArray(int32_t order) {

```

```

int32_t i,j;
for (i = 0; i < count - 1; i++) {
    for (j = i + 1; j < count; j++) {
        if (order == 0) { /*ascending*/
            if (arr[i] > arr[j]) {
                int32_t temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        } else { /*descending*/
            if (arr[i] < arr[j]) {
                int32_t temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

if (order == 0) {
    printf("Array has been sorted ascending.\n");
} else {
    printf("Array has been sorted descending\n");
}

}

void searchElement() {
    if (count == 0) {
        printf("Empty array . No element to search \n");
        return;
    }
}

```

```

    int32_t value;
    printf("Enter element want to search");
    scanf("%d", &value);
    int32_t i;

    for (i = 0; i < count; i++) {
        if (arr[i] == value) {
            printf("element %d searched at position %d.\n", value, i);
            return;
        }
    }

    printf("%d do not find in array\n", value);
}

void printMaxValue() {
    if (count == 0) {
        printf("Empty array . No element to search\n");
        return;
    }

    int32_t max = arr[0];
    int32_t i;
    for (i = 1; i < count; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }

    printf("Max of array= %d.\n", max);
}

```

```

void printMinValue() {
    if (count == 0) {
        printf("Empty array . No element to search\n");
        return;
    }

    int32_t min = arr[0];
    int32_t i;
    for (i = 1; i < count; i++) {
        if (arr[i] < min) {
            min = arr[i];
        }
    }

    printf("Min of Array = %d.\n", min);
}

```

Assignment 3 - Pointer

Viết chương trình cấp phát bộ nhớ động 10 bytes cho 1 con trỏ,
với mẫu hàm được cung cấp như sau:

```
#include <stdint.h>
```

```
#include <stdbool.h>
```

```
#include <stdlib.h>
```

```
bool allocate10Bytes(uint8_t *outPtr);
```

- outPtr: output pointer

- return: true: Nếu cấp phát thành công.

 false: Nếu cấp phát bị lỗi.

Yêu cầu:

Không được thay đổi bất kỳ yêu cầu nào của đề, ví dụ đề yêu cầu viết hàm:

```
bool allocate10Bytes(uint8_t *outPtr);
```


Học viên viết hàm `allocate10bytes(uint8_t *outPtr);`

-> không viết hoa “b” -> vi phạm yêu cầu của đề bài và sẽ bị 0 điểm.

ví dụ:

```
#include <stdint.h>
#include <stdbool.h>
#include <stdlib.h>

bool allocate10Bytes(uint8_t *outPtr)
{
    ???? --> malloc
}

int main()
{
    uint8_t *Ptr_Output = NULL;
    status = allocate10Bytes();
    Ptr_Output[9] = 10;
}
```

Phương pháp giải

```
bool allocate10Bytes(uint8_t *outPtr){
    uint8_t** outPtr1 = (uint8_t**)(outPtr);
    *outPtr1 = malloc(10 * sizeof(uint8_t));

    if (*outPtr1 == NULL) {
        return false;
    }

    return true;
}

int main(void) {
    uint8_t* Ptr_Output = NULL;
```

```

bool status = allocate10Bytes((uint8_t*)&Ptr_Output);

if (status)
Ptr_Output[9] = 10;

free(Ptr_Output);
return 0;
}

```

Assignment 4 - Macro

Viết macro, ví dụ sử dụng demo

set bit thứ n trong 1 thanh ghi có địa chỉ cho trước

REG_BIT_SET32(add,n)

set 1 bit đơn lẻ, bao toàn những bit còn lại

REG_BIT_CLEAR32(add,n)

clear 1 bit đơn lẻ, bao toàn những bit còn lại

SWAP_BYTE32(add)

convert big endian-little endian

SWAP_BYTE16(add)

convert big endian-little endian

add: địa chỉ

n: vị trí từ phải sang trái, từ bit trọng số thấp sang trọng số cao, vị trí so sánh với số 0

ví dụ

```
uint32_t a=0x12345678;
```

```
SWAP_BYTE32(&a);
```

```
/*a = 0x78563412*/
```

```
uint16_t b = 0x1234;
```

```
SWAP_BYTE32(&b);
```

```
/*b = 0x3412*/
```

```
uint32_t b = 0;
REG_BIT_SET32(&b,2);
/* b = 4*/
```

Phương pháp giải

```
#define REG_BIT_SET32(add, n) ((*add) |= (1 << (n)))
#define REG_BIT_CLEAR32(add, n) ((*add) &= ~(1 << (n)))
#define SWAP_BYTE32(add) ((*add) = (((*add) & 0xFF000000) >> 24) | \
                                   (((*add) & 0x00FF0000) >> 8) | \
                                   (((*add) & 0x0000FF00) << 8) | \
                                   (((*add) & 0x000000FF) << 24))

#define SWAP_BYTE16(add) (*add) = (((*add) & 0x00FF) << 8) | \
                                   (((*add) & 0xFF00) >> 8))
```

Assignment 5 - Dynamic Linked list

Assignment 5

- Write a program for managing the students which contains below info:

Student's name

Student's ID

Math score

- Requirements:

Using single linked list for management.

Implement the function to add, remove, find a student information from list by student's ID

Implement a console menu for using feature add, remove, find

Follow to coding convention as other assignments.

Phương pháp giải

```

bool isIdDuplicate(unsigned int id) {
    Student* current = head;
    while (current != NULL) {
        if (current->id == id) {
            return true;
        }
        current = current->next;
    }
    return false;
}

/*****
*FUNCTION ADD STUDENT
*****/

void addStudent() {
    Student* newStudent = (Student*)malloc(sizeof(Student));

    if (newStudent == NULL) {
        printf("Memory allocation error\n");
        return;
    }

    printf("Enter name: ");
    scanf(" %[^\\n]s", newStudent->name);

    printf("Enter ID: ");
    scanf("%u", &newStudent->id);

    if (isIdDuplicate(newStudent->id)) {
        printf("ID already exists. Cannot add student.\n");
        free(newStudent);
        /*Free the allocated memory since we are not adding the student.*/
        return;
    }
}

```

```

    }

    printf("Enter math score: ");
    scanf("%hu", &newStudent->math_score);

    newStudent->next = head;
    head = newStudent;

    printf("Added successfully.\n");
}

/*****
FUNCTION REMOVE STUDENT
*****/
void removeStudent(uint32_t id) {
    Student* current = head;
    Student* prev = NULL;

    while (current != NULL) {
        if (current->id == id) {
            if (prev == NULL) {
                head = current->next;
            } else {
                prev->next = current->next;
            }
            free(current);
            printf("ID %u removed successfully.\n", id);
            return;
        }
        prev = current;
        current = current->next;
    }
}

```

```

    printf("ID %u not found.\n", id);
}

/*****
*FUNCTION FIND STUDENT
*****/

void findStudent(uint32_t id) {
    Student* current = head;

    while (current != NULL) {
        if (current->id == id) {
            printf("Student Information:\n");

            printf("Name: %s\n", current->name);
            printf("ID: %u\n", current->id);
            printf("Math Score: %hu\n", current->math_score);
            return;
        }
        current = current->next;
    }

    printf("ID %u not found.\n", id);
}

```

Assignment 6 - Pointer advance - Static linked list

Pointer advance assignment

Cho một mảng gồm 20 bytes “uint8_t assignment4[20] = {0xFF, 0xFF,...,0xFF}” dùng để lưu trữ các giá trị từ 0 đến 100.

Dùng danh sách liên kết đơn để quản lý các phần tử trong mảng theo giá trị từ thấp đến cao và viết “menu” chương trình cho phép người dùng:

+ Nhập 1 giá trị vào 1 phần tử chỉ định của mảng trên (Dùng phím 1 để chọn option này)

Vị trí từ 0 đến 19. Báo lỗi nếu:

Vị trí người dùng chọn ngoài khoảng 0 đến 19

Vị trí đã được nhập rồi.

Giá trị từ 0 đến 100. Báo lỗi nếu:

Giá trị người dùng chọn ngoài khoảng 0 đến 100

Giá trị người dùng chọn đã tồn tại ở trong mảng.

+ Xóa 1 giá trị nào đó trong mảng trên (Dùng phím 2 để chọn option này)

Sau khi chọn option này, người dùng sẽ chọn 1 giá trị từ 0 đến 100 để xóa, ví dụ người dùng chọn 50:

Nếu giá trị này nằm trong mảng trên thì xóa phần tử này khỏi mảng.

Nếu giá trị này không nằm trong mảng, báo lỗi không tồn tại.

Nếu giá trị người dùng chọn ngoài khoảng từ 0 đến 100, báo lỗi giá trị chọn không đúng.

+ In giá trị các phần tử được nhập vào mảng (Dùng phím 3 để chọn option này)

Sau khi chọn option này, người dùng chọn

1: In các giá trị đã được nhập của mảng khi chưa sắp xếp từ thấp đến cao.

2: In các giá trị đã được nhập của mảng sau khi đã sắp xếp (dùng danh sách liên kết đơn theo yêu cầu ban đầu).

Kết thúc chương trình (Dùng phím 4 để chọn option này).

Chú ý: Không được dùng các hàm cấp phát bộ nhớ động.

Phương pháp giải :

```
void addElement(uint8_t *array, uint8_t position, uint8_t value) {
    int32_t i;
    if (position < 0 || position > 19) {
        printf("Invalid position.\n");
        return;
    }

    if (array[position] != 0xFF) {
        printf("Position is already has value.\n");
        return;
    }
}
```

```

}

if (value < 0 || value > 100) {
    printf("Value Invalid.\n");
    return;
}

for (i = 0; i < 20; i++) {
    if (array[i] == value) {
        printf("Value already exists.\n");
        return;
    }
}

array[position] = value;

printf("Value %d has been added at position %d \n", value, position);
}

void deleteElement(uint8_t *array, uint8_t value) {
    int32_t found = 0;
    int32_t i;
    if (value < 0 || value > 100) {
        printf("Value Invalid.\n");
        return;
    }

    for (i = 0; i < 20; i++) {
        if (array[i] == value) {
            array[i] = 0xFF;
            found = 1;
        }
    }
}

```



```

    }

    if (found) {
        printf("Value %d has been deleted.\n", value);
    } else {
        printf("Value does not exist\n");
    }
}

void printArray(uint8_t *array, uint8_t option) {
    if (option == 1) {
        printf("Value unsort\n");
        int32_t i;
        for (i = 0; i < 20; i++) {
            if (array[i] != 0xFF) {
                printf("%hhu ", array[i]);
            }
        }
        printf("\n");
    } else if (option == 2) {
        printf("Value sorted\n");
        // Use bubble sort
        int32_t i, j;
        for (i = 0; i < 19; i++) {
            for (j = 0; j < 19 - i; j++) {
                if (array[j] > array[j + 1]) {
                    uint8_t temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }
    }
}

```

```

        for (i = 0; i < 20; i++) {
            if (array[i] != 0xFF) {
                printf("%hhu ", array[i]);
            }
        }
        printf("\n");
    } else {
        printf("Error option.\n");
    }
}

```

Mock Project C

Mock Project

Viết chương trình parse file định dạng srecord

- Kiểm tra lỗi và in ra màn hình nếu có (dòng bị lỗi và chi tiết lỗi)
- In ra file Output.txt với nội dung gồm các dòng, mỗi dòng gồm số thứ tự, Address và Data, cách nhau bởi khoảng trắng (space)

Link ref: [https://en.wikipedia.org/wiki/SREC_\(file_format\)](https://en.wikipedia.org/wiki/SREC_(file_format))

Phương pháp giải :

```

/*****
FUNCTION
*****/

static int8_t hexToDec(int8_t temp);

static int8_t hexToDec(int8_t temp){
    if (temp >= '0' && temp <= '9'){
        temp = temp - '0';
    }
    else{
        temp = temp - 'A' + 10;
    }
}

```

```

    return temp;
}

/*****
FUNCTION read_line
*****/
void read_line(int8_t *line, FILE *file, uint8_t *length){
    int8_t temp = 0;
    uint8_t index = 0;

    *length = 0;
    temp = fgetc(file);

    while (temp != '\n' && temp != EOF){
        line[index] = temp;
        index++;
        ++*length;
        temp = fgetc(file);
    }

    line[index] = 0;
}

/*****
FUNCTION Check_S
*****/
int8_t Check_S(uint8_t line){
    int8_t save_er = 0;

    if (line != 'S'){
        save_er = 1;
    }

    return save_er;
}

```

```

}

/*****
FUNCTION Check_Type
*****/

int8_t Check_Type(uint8_t type){
    int8_t save_er = 0;

    if (type < '0' || type > '9')
    {
        save_er = 2;
    }

    return save_er;
}

/*****
FUNCTION Check_Hexa
*****/

int8_t Check_Hexa(int8_t *temp){
    int8_t save_er = 0;
    ++temp;

    while (*temp != 0){
        if (hexToDec(*temp) < 0 || hexToDec(*temp) > 15){
            save_er = 3;
        }
        ++temp;
    }

    return save_er;
}

/*****
FUNCTION Check_Byte_Count
*****/

```

```

int8_t Check_Byte_Count(int8_t *str){
    int8_t save_er = 0;
    uint8_t length = 0;
    uint8_t bytes = 0;

    length = strlen(str) - 4;
    bytes = num2(hexToDec(str[2]), hexToDec(str[3]));

    if (bytes != (length / 2)){
        save_er = 4;
    }

    return save_er;
}

/*****
FUNCTION Check_Sum
*****/
int8_t Check_Sum(int8_t *str){
    uint8_t lastByte = 0;
    uint8_t length = 0;
    int8_t save_er = 0;
    uint8_t sum = 0;
    uint8_t index = 0;

    length = strlen(str);
    lastByte = ~(num2(hexToDec(str[length - 2]), hexToDec(str[length - 1])));

    for (index = 2; index < length - 2; index += 2){
        sum += num2(hexToDec(str[index]), hexToDec(str[index + 1]));
    }

    if (sum != lastByte){
        save_er = 5;
    }
}

```

```

    }

    return save_er;
}

/*****
FUNCTION Check_Line_Count
*****/
int8_t Check_Line_Count(int8_t *str, uint16_t count){
    int8_t save_er = 0;
    uint16_t checkLine = 0;

    checkLine = num2(hexToDec(str[4]), hexToDec(str[5]));
    if (count != checkLine){
        save_er = 6;
    }

    return save_er;
}

/*****
FUNCTION Check_Terminate
*****/
int8_t Check_Terminate(int8_t temp, uint8_t type){
    int8_t save_er = 0;

    if (temp != type){
        save_er = 7;
    }

    return save_er;
}

/*****
FUNCTION print_data

```

```

*****/
void print_data(int8_t *str, int8_t type, uint8_t length, FILE *file){
    uint8_t index = 0;
    if (type == '3'){
        for (index = 4; index < length - 2; index++){
            if (index == 12){
                fprintf(file, " : Address - DATA : ");
                printf(" : Address - DATA : ");
            }

            fprintf(file, "%c", str[index]);
            printf("%c", str[index]);
        }
    }
    else if (type == '2'){
        for (index = 4; index < length - 2; index++){
            if (index == 10){
                fprintf(file, " ");
                printf(" ");
            }

            fprintf(file, "%c", str[index]);
            printf("%c", str[index]);
        }
    }
    else if (type == '1'){
        for (index = 4; index < length - 2; index++){
            if (index == 8)
            {
                fprintf(file, " ");
                printf(" ");
            }

            fprintf(file, "%c", str[index]);

```

```
        printf("%c", str[index]);  
    }  
}  
}
```

2.3. Khóa học MCP

Ass1

Viết chương trình nháy LED đỏ bằng nút SW1, SW2.

- Mode 1: Led sáng dần mượt mà rồi tắt dần rồi lại sáng dần theo chu kì 2s, trong 2 giây đó giây đầu

led sáng dần, giây sau led tắt dần.

- Mode 2: Nhấn nút SW1 để tăng độ sáng của led, từ 0%, tăng lên 25%, 50%, 75%, 100%, rồi về 0%...

- Chuyển giữa 2 mode bằng cách nhấn nút SW2

Note:

- Nên viết hàm PWM có tham số dutyCycle
- Sử dụng state machine

Phương pháp giải :

```
int main(void)  
{  
    initLed();  
    initButtons();  
  
    while (1)  
    {  
        if (currentMode == MODE1)  
        {  
            mode1();  
        }  
    }  
}
```



```

        else if (currentMode == MODE2)
        {
            mode2();
        }
    }
}

void initLed()
{
    /* Enable clock for PORTE */
    SIM->SCGC5 |= (1 << 13);

    /* Initialize the RED LED (PTE29) */
    /* Set the PTE29 pin multiplexer to GPIO mode */
    PORTE->PCR[29] |= PORT_PCR_MUX(1);
    /* Set the pin's direction to output */
    GPIOE->PDDR |= RED_LED_PIN;
    /* Set the initial output state to low (LED off) */
    GPIOE->PCOR |= RED_LED_PIN;
}

void initButtons()
{
    /* Enable clock for PORTC */
    SIM->SCGC5 |= (1 << 11);

    /* Set the SW1 and SW2 pins (PTC3 and PTC12) as input */
    GPIOC->PDDR &= ~(SW1_PIN | SW2_PIN);

    /* Enable the internal pull-up resistors for SW1 and SW2 */
    PORTC->PCR[3] |= PORT_PCR_MUX(1) | PORT_PCR_PE_MASK | PORT_PCR_PS_MASK;
    PORTC->PCR[12] |= PORT_PCR_MUX(1) | PORT_PCR_PE_MASK | PORT_PCR_PS_MASK;
}

```

```

    /* Enable interrupt for SW2 (PTC12) */
    NVIC_EnableIRQ(PORTA_IRQn); // Corrected: Assign the interrupt to PORTC
    PORTC->PCR[12] |= PORT_PCR_IRQC(0x9); // Falling edge interrupt
}

void delay()
{
    uint32_t i;
    for (i = 0; i < DELAY_CNT; i++)
    {
        __asm("nop");
    }
}

void mode1()
{
    while (currentMode == MODE1)
    {
        /* Fade in the LED */
        for (brightnessLevel = 0; brightnessLevel <= 100; brightnessLevel += 5)
        {
            GPIOE->PCOR = RED_LED_PIN; // Turn off the LED
            delay();

            // Calculate delay time based on brightness level
            uint32_t delayTime = (brightnessLevel * DELAY_CNT) / 100;
            for (uint32_t i = 0; i < delayTime; i++)
            {
                __asm("nop");
            }
        }
    }
}

```

```

GPIOE->PSOR = RED_LED_PIN; // Turn on the LED
delay();

/* Calculate delay time based on brightness level */
delayTime = ((100 - brightnessLevel) * DELAY_CNT) / 100;
for (uint32_t i = 0; i < delayTime; i++)
{
    __asm("nop");
}

/* Delay for 2 seconds */
for (int i = 0; i < 4; i++)
{
    delay();
}

// Fade out the LED
for (brightnessLevel = 100; brightnessLevel >= 0; brightnessLevel -= 5)
{
    GPIOE->PCOR = RED_LED_PIN; // Turn off the LED
    delay();

    // Calculate delay time based on brightness level
    uint32_t delayTime = (brightnessLevel * DELAY_CNT) / 100;
    for (uint32_t i = 0; i < delayTime; i++)
    {
        __asm("nop");
    }

    GPIOE->PSOR = RED_LED_PIN; // Turn on the LED
    delay();
}

```

```

        /* Calculate delay time based on brightness level */
        delayTime = ((100 - brightnessLevel) * DELAY_CNT) / 100;
        for (uint32_t i = 0; i < delayTime; i++)
        {
            __asm("nop");
        }
    }

    // Delay for 2 seconds
    for (int i = 0; i < 4; i++)
    {
        delay();
    }
}

void mode2()
{
    while (currentMode == MODE2)
    {
        if ((GPIOC->PDIR & SW1_PIN) == 0)
        {
            /* SW1 is pressed, increase brightness level */
            brightnessLevel += 25;
            if (brightnessLevel > 100)
            {
                brightnessLevel = 0;
            }

            /* Debounce delay to avoid rapid changes when holding the button */
            delay();
        }
    }
}

```

```

    }
}

void PORTC_IRQHandler()
{
    /* Check if SW2 is pressed (Mode switch) */
    if ((GPIOC->PDIR & SW2_PIN) == 0)
    {
        /* SW2 is pressed, change the mode */
        if (currentMode == MODE1)
        {
            currentMode = MODE2;
        }
        else
        {
            currentMode = MODE1;
            /* Reset brightness level when switching back to Mode 1 */
            brightnessLevel = 0;
        }

        /* Clear the interrupt by writing 1 to the corresponding bit in the ISFR */
        PORTC->ISFR |= SW2_PIN;

        /* Debounce delay to avoid rapid mode switching */
        delay();
    }
}

```

Ass2

Write an example application to blink a led RED by using a system timer with frequency 2Hz (SysTick) --> Max 8 point

Advance: Use interrupt for button to:

Press switch 1 to turn on led GREEN and press switch 2 to turn of led GREEN. --> Max 10 point

Phương pháp giải :

```
#define RED_LED_PIN      (1 << 29)
#define GREEN_LED_PIN    (1 << 5)
#define SW1_PIN          (1 << 3)
#define SW2_PIN          (1 << 12)
#define BLINK_INTERVAL   (500) // Blink interval in milliseconds (2Hz)

volatile uint32_t g_msTicks = 0; /* Counter for milliseconds */
/*****
 * FUNCTION
 *****/
void SysTick_Handler(void) {
    g_msTicks++;
}

void initSysTick(void) {
    /* Initialize System Tick Timer to generate interrupts every 1 ms */
    SysTick_Config(SystemCoreClock / 1000);
}

void initLed(void) {
    // Enable clock for PORTE & PORTD
    SIM->SCGC5 |= (1 << 12 | 1 << 13);

    /* Initialize the RED LED (PTE29) */
    PORTE->PCR[29] |= PORT_PCR_MUX(1);
    GPIOE->PDDR |= RED_LED_PIN;
```

```

// Initialize the Green LED (PTD5)
PORTD->PCR[5] |= PORT_PCR_MUX(1);
GPIOB->PDDR |= GREEN_LED_PIN;

/* Turn off LEDs initially */
GPIOE->PSOR |= RED_LED_PIN;
GPIOB->PSOR |= GREEN_LED_PIN;
}

void initButtons(void) {
    // Enable clock for PORTC
    SIM->SCGC5 |= (1 << 11);

    /* Configure SW1 and SW2 pins as input */
    PORTC->PCR[3] |= PORT_PCR_MUX(1) | PORT_PCR_PE_MASK | PORT_PCR_PS_MASK;
    PORTC->PCR[12] |= PORT_PCR_MUX(1) | PORT_PCR_PE_MASK | PORT_PCR_PS_MASK;
}

void delayMs(uint32_t ms) {
    uint32_t startTicks = g_msTicks;
    while ((g_msTicks - startTicks) < ms) {}
}

/*****
 * MAIN
 *****/

int main(void) {
    initLed();
    initButtons();
    initSysTick();

    while (1) {
        /* Blink RED LED with 2Hz frequency */

```

```

GPIOE->PTOR |= RED_LED_PIN;
delayMs(BLINK_INTERVAL);

// Check SW1 for turning on GREEN LED
if ((GPIOC->PDIR & SW1_PIN) == 0) {
    GPIOD->PCOR |= GREEN_LED_PIN; // Turn on GREEN LED
}

/* Check SW2 for turning off GREEN LED */
if ((GPIOC->PDIR & SW2_PIN) == 0) {
    GPIOD->PSOR |= GREEN_LED_PIN; /* Turn off GREEN LED */
}
}
}

```

Mock

Use UART to transfer SREC file from computer to board. If board check SREC correct then write data to flash with address corresponding.

Lecture: Provide module flash for student and guide.

Phương pháp giải :

```

int main() {
    UART_Init();

    FILE *inputFile = fopen("Blink_Led_premitive.s19", "r");
    if (inputFile == NULL){
        printf("\nUnable to open the input file");
        return 1;
    }

    while (!feof(inputFile)){

```



```

uint8_t line[MAX];
uint8_t length = 0;
read_line(line, inputFile, &length);

// Assuming line is a valid SREC line
uint8_t type = line[1];
uint8_t srecData[MAX]; // Store the SREC data here
uint8_t srecLength = 0; // Length of the SREC data
uint32_t srecAddress = FLASH_START_ADDRESS;

// Extract and process the SREC data (skip error checking for now)
for (uint8_t index = 4; index < length - 2; index += 2){
    srecData[srecLength] = (hexToDec(line[index]) << 4) | hexToDec(line[index
+ 1]);
    srecLength++;
}

// Check for record types 1, 2, and 3
if (type == '1' || type == '2' || type == '3'){
    // Process the data and write to Flash memory if needed
    if (type == '3'){
        // Assuming you have the starting address for Flash memory
(srecAddress)
        // and the SREC data (srecData) extracted above
        WriteDataToFlash(&srecAddress, (uint32_t *)srecData, srecLength / 4);
    }
}

fclose(inputFile);

return 0;
}

```

CHƯƠNG 3: NHẬN XÉT, ĐỀ XUẤT

3.1. Ưu điểm

Đã có hiểu biết bản về lập trình C Tiếp thu kiến thức mới phần MCP khá nhanh Có khả tự học phát triển thêm Nộp tập đúng hạn, đầy đủ Tham gia học đầy đủ, đúng giờ

3.2. Nhược điểm

Chưa thể tiếp thu hết kiến thức mà công ty truyền đạt

CHƯƠNG 4: KẾT LUẬN

FSOFT cung cấp môi trường lý tưởng cho cả sinh viên trường và những sinh viên muốn làm thêm để tích lũy kinh nghiệm. Với bản chất làm việc đầy tính động, khoa học và chuyên nghiệp, FSOFT góp phần phát triển tốt năng lực của từng thành viên. Để đáp ứng yêu cầu của khách hàng và nắm bắt thay đổi công nghệ, nhân viên FSOFT luôn được đào tạo mới và phát triển không ngừng. Mặc dù thời gian thực tập khá ngắn nhưng đã đủ để để lại những ấn tượng sâu sắc trong tâm trí của sinh viên.

Ấn tượng đầu tiên khi bước chân vào FSOFT là môi trường làm việc tươi mới, không chỉ thiên nhiên mà còn cả người. Tinh thần thân thiện và nhiệt tình của nhân viên tạo nên môi trường vui vẻ, sôi động. Được xây dựng và phát triển qua thời gian, FSOFT tự hào về thành tích đạt được, một thành quả đến từ sự nỗ lực không ngừng của đội ngũ tận tâm.

Đặc biệt, công ty sở hữu môi trường làm việc hiện đại với không gian chuyên nghiệp. Không khí làm việc thể hiện qua giao lưu giữa sinh viên và nhân viên, tạo nên sự cởi mở và vui vẻ, điều kiện tốt cho những kỹ sư tương lai phát triển sự sáng tạo và đam mê.

Buổi thực tập gần đây đã mang lại những trải nghiệm thú vị và kinh nghiệm quý báu cho sinh viên. Đây là nền tảng quan trọng để xác định đam mê đúng hướng và xây dựng kế hoạch chiến lược rõ ràng. Bằng việc học tập chăm chỉ, rèn luyện kiến thức chuyên môn cũng như các kỹ năng mềm và ngoại ngữ, sinh viên có thể tự tin hướng tới tương lai.

Cuối cùng, chúng em xin bày tỏ lòng biết ơn sâu sắc tới FPT Software, Nhà trường và Ban lãnh đạo Viện ĐTVT đã tạo điều kiện cho chúng em tham gia thực tập có ích. Chúng em mong rằng Ban giám hiệu Nhà trường và Ban lãnh đạo Viện ĐTVT sẽ tiếp tục tạo cơ hội cho chúng em tham gia các buổi thực tập thực nghiệm, giúp chúng em nâng cao kiến thức và kỹ năng.